Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра автоматизированных систем управления
Кафедра вычислительной техники

Паспорт
курсовой работы

по дисциплине «Теория формальных языков и компиляторов», 6 семестр

1. **Методика оценки.**

Выполнение курсовой работы (далее – КР) является обязательным видом самостоятельной работы студента по дисциплине, предусмотренным учебным планом.

Основной целью выполнения КР является формирование компетенций и соотнесенных с ними индикаторов по дисциплине «Теория формальных языков и компиляторов», 6 посредством закрепления, углубления и обобщения знаний, полученных студентами за время теоретического обучения и прохождения практик, а также выработка навыков самостоятельного применения знаний и навыков для творческого решения конкретных задач. Выполнение курсовой работы должно способствовать подготовке их к решению более сложной задачи - выполнению выпускной квалификационной работы.

Задачами выполнения курсовой работы является овладение студентами рациональными приёмами сбора, обработки, систематизации информации, применения компьютерных технологий в области создания трансляторов, применения нормативно-законодательной базы и умения оценивать эффективность реализуемых проектов и программ в региональной социально-экономической политике.

Тематика КР соответствует профилю (направленности) подготовки, формируется преподавателем в начале семестра и утверждается на заседании кафедры. Количество тем КР достаточно для обеспечения каждого обучающегося.

Выполнение студентами КР начинается с ознакомления с примерной тематикой. Закрепление тем КР за студентами и назначение научных руководителей производится распоряжением заведующего кафедрой и/или утверждается решением кафедры.

Курсовая работа выполняется индивидуально.

Структура курсовой работы:

1. Титульный лист стандартной для НГТУ формы
2. Введение (актуальность, цель, задачи).
3. Лексический анализатор транслятора.
4. Синтаксический анализатор и формирователь АСД или ПФЗ.
5. Семантический анализатор и генератор псевдокода.
6. Интерпретатор псевдокода.
7. Заключение (выводы и рекомендации).
8. Список литературы и источников.

Во введении формулируется цель работы и постановка задачи, обосновывается актуальность.

Глава 1 содержит результаты разработки системы регулярных выражений, определяющих лексику языка и описание построенного по этой системе лексического анализатора в табличном или графовом представлении.

Глава 2 содержит результаты разработки формальной грамматики, определяющей синтаксис заданного языка, описание конечного автомата со стековой памятью в качестве парсера транслятора и алгоритмы преобразования транслируемой программы в промежуточное внутреннее представление – абстрактное синтаксическое дерево (АСД)

или постфиксную форму записи(ПФЗ).

Глава 3 содержит разработанные студентом алгоритмы семантического анализа АСД/ПФЗ и генерации псевдокода для избранной архитектуры виртуальной машины.

Глава 4 содержит описание разработанных студентом алгоритмов интерпретации псевдокода и результаты отладки и тестирования транслятора в целом.

Заключение: изложение общих выводов по изученным использованным методам разработки трансляторов.

Список литературы оформляется в соответствии с библиографическими требованиями в алфавитном порядке и включает от 3 до 9 источников (книг, статей разных авторов, интернет-источников, документов), которые были изучены при выполнении работы.

Требования к оформлению:

Объем КР до 30 страниц машинописного текста формата А4. Шрифт Times New Roman, 12. Нумерация страниц сквозная, в нижней части листа по центру арабскими цифрами. КР должна быть отредактирована, не содержать орфографических, синтаксических и стилевых ошибок.

Законченная курсовая работа предоставляется для проверки в электронном виде в срок, установленный преподавателем. Преподаватель оценивает качество КР с учетом теоретического и практического содержания, достижения ее целей и задач.

Курсовая работа проверяется руководителем работы, который дает письменное заключение по работе — рецензию.

Если при выполнении КР были допущены ошибки, то работа возвращается студенту для исправления выявленных недочетов и затем вновь предоставляется руководителю для проверки. При положительном результате оценивания студент распечатывает работу, передает на кафедру и защищает до сессии в назначенное преподавателем время.

Защита КР проходит публично перед группой студентов.

По результатам защиты студенту выставляется оценка в соответствии с критериями, приведенными в п. 2 настоящего Паспорта.

2. Уровни сформированности компетенций и критерии оценки

В соответствии с балльно-рейтинговой системой НГТУ курсовая работа по дисциплине «Теория формальных языков и компиляторов», выполняемая в 6-м семестре, имеет максимальную оценку 100 баллов.

Курсовая работа выполнен **на продвинутом** уровне, если:

- она выполнена в полном соответствии с заданием, отличается глубиной проработки всех разделов содержательной части, текстовая часть оформлена с соблюдением установленных правил;
- руководитель характеризует деятельность студента положительно (в частности, отмечает его инициативу, самостоятельность, систематичность работы на всех этапах выполнения работы);
- в докладе исчерпывающе, последовательно, четко и логически правильно изложена суть работы и ее основные результаты;
- студент свободно владеет теоретическим материалом, безошибочно применяет его при решении задач, сформулированных в задании;
- в докладе суть работы и ее основные результаты представлены исчерпывающе, последовательно, четко и логически правильно; на все вопросы студент дал обстоятельные и аргументированные ответы, убедительно защищал свою точку зрения;
- компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, по которым выполняется курсовая работа, сформированы в полном объеме.

Оценка за выполнение КР составляет *100-87 баллов.*

Курсовая работа выполнен на **базовом** уровне, если:
- она соответствует заданию, отличается глубиной проработки всех разделов содержательной части, текстовая часть расчетно-пояснительной записки оформлена с соблюдением установленных правил;
- руководитель характеризует деятельность студента положительно, но с незначительными замечаниями;
- в докладе правильно изложена суть работы и ее основные результаты;
- студент достаточно твердо усвоил теоретический материал и может самостоятельно его применять;
- в докладе суть работы и ее основные результаты представлены полно; на все вопросы студент дал ответы, но их полнота и аргументированность недостаточны;
- компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, по которой выполняется курсовая работа, сформированы с небольшими пробелами и соответствуют базовому уровню.

Оценка за выполнение КР составляет *86-73 балла*.

Курсовая работа выполнена **на пороговом** уровне, если:
- выполнена в основном правильно, но без необходимой проработки некоторых разделов;
- в докладе упущены некоторые принципиальные моменты содержательной части работы;
- в докладе представлены суть работы и ее основные результаты, но ответы на вопросы были даны с существенными затруднениями;
- компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, по которой выполняется курсовая работа, сформированы с пробелами и соответствуют пороговому уровню.

Оценка за выполнение КР составляет *72-50 баллов.*

Курсовая работа считается **не выполненной,** если студентом не проработаны важные разделы исследования, допущены принципиальные ошибки, не исправленные после замечаний руководителя курсовой КР. Студент не допущен к защите курсовой работы. Компетенции и соотнесенные с ними индикаторы, закрепленные за дисциплиной, по которой выполняется курсовая работа, не сформированы.
Оценка составляет *менее 49 баллов*.

3. Шкала оценки.

В общей оценке по дисциплине баллы за работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

Курсовая работа по дисциплине считается успешно выполненной, если сумма полученных баллов составляет от 100 до 50 баллов включительно.

Примерный перечень тем курсовой работы

Цель работы: практическое применение теоретических основ проектирования трансляторов с языков программирования; освоение средств автоматизации построения трансляторов; разработка элементов транслятора для учебного языка.

Типовое задание на курсовую работу

Раздел 1.

Описать лексику, синтаксис и семантику заданного варианта языка. Написать несколько простых тестовых программ, содержащих все заданные элементы и управляющие конструкции языка. Вручную выполнить интерпретацию программы. Эти программы использовать впоследствии для проверки элементов разрабатываемого транслятора.

Раздел 2.

2.1. Разработать систему регулярных выражений, определяющую лексику заданного варианта языка. Используя пакет Вебтранслаб, построить автоматную реализацию лексического акцептора на выбранном инструментальном языке. Добиться работоспособности акцептора.

2.2. Разработать совокупность действий, т.е. расширение системы регулярных выражений, реализующую:

a. - поиск в таблицах идентификаторов / констант и пополнение таблиц;
b. - доформирование лексем,
c. и построить лексический анализатор. Убедиться в правильности его работы.

Раздел 3.

3.1. Разработать формальную грамматику класса LL(1), определяющую синтаксис заданного языка. Используя пакет Вебтранслаб, построить какую-либо автоматную реализацию нисходящего синтаксического акцептора, добиться его работоспособности.

Или:

3.2. Разработать формальную грамматику класса не выше, чем LALR(1). Используя пакет Вебтранслаб, построить автоматную реализацию восходящего синтаксического акцептора, добиться его работоспособности.

3.3. Разработать расширение построенного синтаксического акцептора для преобразования входной последовательности лексем в постфиксную форму записи или в абстрактное синтаксическое дерево.

Раздел 4.

Разработать семантический анализатор и преобразователь ПФЗ или АСД в псевдокод. Формат псевдокода определяется последней цифрой варианта задания.

Раздел 5.

Разработать генератор объектного кода или интерпретатор псевдокода.

II Определение учебного языка, варианты заданий.

Лексика, синтаксис и семантика учебного языка основаны на языках типа Java и C#. Любые конструкции, не оговоренные явно в задании, можно определять самостоятельно.

II.1 Идентификаторы

1. <б><пЦ><пБ> (например: v5Zx, i7gh, p197ART, …)
2. $<пЦ><пБ> (например: $12aD, $7field, $0Member, …)
3. <пБ><пЦ> (например: v5Z, i7, ART197, …)
4. <бБ><пЦ><бБ> (например: x123y, N1n, a0B, …)
5. _<пБЦ>_ (например: _K_, _37_, _out_, …)

II.2 Константы

1. целые по основаниям 2, 8 и 10; вещественные; строковые и символьные
2. целые по основаниям 4, 10 и 16; вещественные; строковые и символьные
3. целые по основаниям 4, 8 и 10; вещественные; строковые и символьные
4. целые по основанию 10; вещественные и экспоненциальные; символьные
5. целые по основаниям 2, 4 и 10; вещественные; строковые и символьные

Правильными словами языка являются также знаки операций и все последовательности символов, выделенные жирным шрифтом в разделах «Синтаксис» и

«Семантика» (за исключением символов «**[**» и «**]**», см. раздел «Обозначения»).

Синтаксис

II.3 Оператор присваивания

1. **let** <И> **on** <В> **;**
2. <И> **:=** <В> **;**
3. <И> **<–** <В> **;**
4. **put**<В> **to** <И> **;**
5. <И> **set** <В> **;**

II.4 Условный оператор

1. **at** <В> **do** <ОБ> **[or do** <ОБ> **]**
2. **if (** <В> **) then** <ОБ> **[not**<ОБ> **]**
3. **when** <В> <ОБ> **[other** <ОБ> **]**
4. **? (** <В> **)** <ОБ> **[?: (** <В> **)** <ОБ> **] [:** <ОБ> **]**
5. **by (** <В> **)** <ОБ> **[else** <ОБ> **]**

II.5 Оператор цикла

Для выхода из цикла его тело ОБ (оператор или блок) может содержать оператор, указанный в дополнительных квадратных скобках; таких операторов в одном цикле может быть несколько

1. **exec** <ОБ> **with** <И> **from** <К> **to** <К> **[step** <К> **]** **[exit;]**
2. **cycle ([** <О> **] ; [** <В> **] ; [** <О> **])** <ОБ> **[out;]**
3. **while (** <Л> **) do** <ОБ> **[break;]**
4. **loop** <ОБ> **until [** <В> **)** **[endloop;]**
5. **foreach (** <И> **in** <К> **:** <К> **)** <ОБ> **[end;]**

II.6 Оператор переключателя

Для выхода из переключателя последовательность операторов/блоков любой ветки может завершаться оператором, указанным в квадратных скобках

1. **choice** <В> **option** <К> **:** <ПО> **[fin;] [option …] … [nooption** <ПО> **] end**
2. **case** <В> **{ when** <К> **then** <ПО> **[exit;] [when…] … [otherwise** <ПО> **] }**
3. **select** <В> **case (** <К> **)** <ПО> **[break;] [case …] … [case ()** <ПО> **] end**
4. **??** <В> **{ ?=** <К> **:** <ПО> **[gout;] [?= …] … [?~ :** <ПО> **] }**
5. **switch** <В> **{ by** <К> **do** <ПО> **[off;] [by …] … [any do** <ПО> **] }**

II.7 Объявление функций

1. **[** <Тип> **]** <И> **([**<АргЛист>**])** <Тело> (например int _Mrg_(float _Af_) begin…end)
2. **=> [as** <Тип> **] ([** <АргЛист> **])** <Тело> (например i1let = => as char () {…})
3. <И> **([** <АргЛист> **]) [ret** <Тип> **]** <Тело> (например $3Pi() ret long {…})
4. **function ([** <АргЛист> **])** <Тело> (например mF = function(int fArg) { … })
5. **[** <Тип> **] ([** <АргЛист> **])** <Тело> (например $0var = int (double $5t) start … end)

Семантика

II.8 Объявления типов

1. **Обязательны, строгая типизация (неявные преобразования запрещены)**
2. **Отсутствуют (явные преобразования возможны)**
3. **Обязательны, нестрогая типизация (возможны неявные преобразования)**
4. **Могут присутствовать, но не обязательны**
5. **Отсутствуют (явные преобразования запрещены)**

II.9 Формат псевдокода

1. Тетрады: <Код><Оп><Оп><Р>
2. Триады: <Код><Оп><Р>
3. Пентады: <Метка><Код><Оп><Оп><Р>
4. Триады: <Код><Оп><Оп>
5. Тетрады: <Метка><Код><Оп><Оп>

Обозначения:

[...] – квадратные скобки охватывают необязательную часть конструкции; в объявлениях функций обязательность присутствия типа возвращаемого значения определяется восьмой цифрой варианта задания; примеры в объявлениях функций могут

содержать операторы присваивания, отличающиеся от заданных, ориентироваться следует на третью цифру варианта задания;

… – предшествующая часть конструкции повторяется произвольное количество раз;

< > – сокращения:

<б> – одна маленькая буква;
<Б> – одна большая буква;
<бБ> – одна любая буква;
<бЦ> – одна маленькая буква или одна цифра;
<Ц> – одна цифра;
<пБ> – непустая последовательность букв;
<пЦ> – последовательность цифр длины от 1 до 4;
<пБЦ> – возможно пустая последовательность букв и/или цифр
<И> – идентификатор (имя переменной или объекта);
<К> – константа;
<В> – выражение;
<ОБ> – одиночный оператор или блок операторов;
<О> – одиночный оператор;
<ПО> – последовательность из одного или нескольких операторов / блоков;
<Тип> – ключевое слово типа данных;
<Тело> – блок операторов (например { … } или begin … end);
<АргЛист> – список формальных аргументов функции;
<Код> – поле кода операции;
<Метка> – поле метки;
<Оп> – поле наименования операнда;
<Р> – поле наименования результата.

Любые конструкции, явно не определенные в этом задании (например – массивы, или указатели, или внутренние классы/интерфейсы/…), можно вводить в язык самостоятельно, однако делать этого не рекомендуется, сложность работы и без того высока.

4. Примерный перечень вопросов к защите курсового проекта (работы).

1. Как задать одну группу слов несколькими разными регулярными выражениями?
2. Как включать действия в системы регулярных выражений?
3. Какие операции используются в языке регулярных выражений?
4. Что такое мультиавтоматный лексический акцептор и для чего он может понадобиться?
5. Что такое контекстно-свободная грамматика?
6. Что такое дерево грамматического разбора и каковы его свойства?
7. Как используются регулярные выражения в грамматиках?
8. Как определить пригодность грамматики к выбранному способу синтаксического анализа – нисходящему или восходящему?
9. Что такое расширение синтаксического акцептора?
10. Как включать действия в формальную грамматику?
11. Что такое отношения предшествования и последования, как и для чего они используются построителем синтаксических анализаторов?
12. Что такое аннулируемые, бесплодные и недостижимые нетерминалы грамматики, как информация о таких символах используется при преобразовании грамматики в конечный автомат со стековой памятью?
13. Какими свойствами обладает постфиксная форма записи?
14. Что такое абстрактное синтаксическое дерево?
15. Как дерево грамматического разбора преобразуется в абстрактное синтаксическое дерево?
16. Что такое множество выбора правила грамматики?
17. Что такое рекурсивный спуск?

18. Какие операции выполняет восходящий синтаксический анализатор?
19. Какие виды псевдокода могут быть использованы в трансляторах?
20. В чем состоит отличие компилятора от интерпретатора?
21. Как в трансляторах реализуется контроль типов данных?
22. Что такое ассоциация наименования объекта программы?
23. Что такое активация функции и вызывающая последовательность?