

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Рабочая программа дисциплины

Методы компиляции

по направлению подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки:
Прикладная математика и инженерия цифровых проектов

Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2025

СОГЛАСОВАНО:
Руководитель ОП
Д.Д. Даммер

Председатель УМК
С.П. Сущенко

Томск – 2025

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.1 Обладает навыками объектно-ориентированного программирования для решения прикладных задач в профессиональной деятельности.

ИОПК-2.2 Проявляет навыки использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.

ИОПК-2.3 Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи.

ИОПК-2.4 Демонстрирует умение адаптировать существующие математические методы для решения конкретной прикладной задачи.

2. Задачи освоения дисциплины

- Изучение формальных грамматик и их классификация
- Изучение основных методов реализации блоков лексического и синтаксического анализа
- Изучение методов детерминированного синтаксического анализа на основе восходящей и нисходящей стратегий
- Изучение методов оптимизация программ, описанных в виде промежуточного представления

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы. Дисциплина входит в модуль «Разработка программного обеспечения».

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Шестой семестр, зачет с оценкой

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: «Основы программирования», «Алгоритмы и структуры данных», «Дискретная математика».

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 3 з.е., 108 часов, из которых:

-лекции: 32 ч.

-лабораторные: 32 ч.

в том числе практическая подготовка: 32 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Элементы теории формальных грамматик

Задачи и основные этапы трансляции. Определение формальной грамматики и формального языка. Определение вывода, разбора. Деревья вывода. Метаязык БНФ. Основная задача синтаксического анализа. Классификация формальных грамматик.

Типы контекстно-свободных грамматик. Допустимые преобразования грамматик. Критерий порождения грамматикой бесконечного языка. Преобразование укорачивающей грамматики в неукорачивающую. Общая модель распознавателя.

Недетерминированные методы синтаксического анализа.

Тема 2. Лексический анализ

Определение сканера. Две стратегии лексического анализа. Автоматные языки и их свойства. Диаграмма состояний автоматной грамматики. Распознаватель автоматных языков. Программирование сканера. Блок лексического анализа. Конструирование сканера.

Тема 3. Методы детерминированной синтаксического анализа

Метод простого предшествования.

Методы расширенного и операторного предшествования.

LR(k)-грамматики, LR(1)-анализатор

LL(k)-грамматики, LL(1)-анализатор

Расширенная БНФ. Метод рекурсивного спуска. Метод Кока-Янгера-Касами

Тема 4. Атрибутный анализ

Понятие идентификации. Простейшая реализация идентификации. Учёт областей действия. Атрибутная индукция

Тема 5. Оптимизация программ

Основные оптимизирующие преобразования. Оптимизация на линейных участках. Свёртка констант. Исключение лишних операций. Оптимизация циклов.

Тема 6. Генерация кода

Польская инверсная запись арифметического выражения. Трансляция на Ассемблер. Динамические деревья. Трансляция условных операторов и циклов.

9. Текущий контроль по дисциплине

Текущий контроль по дисциплине проводится путем контроля посещаемости, контроля выполнения лабораторных работ, опросов по лекционному материалу и фиксируется в форме контрольной точки не менее одного раза в семестр.

10. Порядок проведения и критерии оценивания промежуточной аттестации

Зачет с оценкой в шестом семестре проводится в письменной форме по билетам. Билет содержит два теоретических вопроса, и одну задачу. Вопросы и задания билета проверяют знания и умения по индикаторам всех компетенций дисциплины.

Продолжительность зачета 1,5 часа.

Примерный перечень теоретических вопросов

1. Задачи и этапы трансляции. Типы трансляторов.
2. Формальные грамматики и языки: грамматика, вывод, язык, разбор. Метаязык БНФ.
3. Способы задания выводов и разборов. Деревья вывода/разбора.
4. Классификация грамматик и языков по Хомскому.
5. Две стратегии распознавания контекстно-свободных языков. Неформальное описание нисходящей стратегии.

6. Две стратегии распознавания контекстно-свободных языков. Неформальное описание восходящей стратегии.
7. Допустимые преобразования контекстно-свободных грамматик. Типы контекстно-свободных грамматик.
8. Необходимое и достаточное условие порождения грамматикой бесконечного языка.
9. Преобразование укорачивающей контекстно-свободной грамматики в контекстно-свободную.
10. Назначение сканера, две стратегии использования. Теорема о преобразовании линейной грамматики в автоматную.
11. Диаграмма состояний лексем. Построение диаграммы состояний по правилам грамматики класса 3. Пример для лексемы "текстовая константа"
12. Диаграмма состояний лексем. Построение диаграммы состояний по правилам грамматики класса 3. Пример для лексемы "целочисленная константа"
13. Блок лексического анализа: дескрипторы, лексическая свёртка, таблицы.
14. Блок лексического анализа: задачи, структура, результат работы.
15. Метод простого предшествования: определение грамматики предшествования.
16. Метод простого предшествования: алгоритм работы распознавателя.
17. Метод простого предшествования: определение множеств $L(U)$ и $R(U)$, алгоритм их построения.
18. Метод простого предшествования: построение матрицы предшествования.
19. Метод простого предшествования: функции предшествования. Алгоритм Флойда построения функций предшествования.
20. Метод простого предшествования: функции предшествования. Построение функций предшествования по графу линеаризации.
21. Метод простого предшествования: структура транслятора.
22. Метод операторного предшествования: определение грамматики.
23. Метод операторного предшествования: первичная фраза, работа распознавателя.
24. Метод операторного предшествования: определение множеств $L_t(U)$ и $R_t(U)$, алгоритм их построения.
25. Метод операторного предшествования: матрица операторного предшествования, функции операторного предшествования.
26. LR(k)-грамматики. Определение.
27. LR(1)-распознаватель, его структура и работа.
28. LL(k)-грамматики. Определение. Иллюстрация на деревьях вывода.
29. 1-предсказывающий алгоритм разбора: структура и работа.
30. Метод рекурсивного спуска: структура анализатора и его работа.
31. Факторизация грамматики. Расширенная БНФ. Примеры.
32. Метод Кока-Янгера-Касами: алгоритм построения разбора.
33. Метод Кока-Янгера-Касами: алгоритм формирования левого вывода.
34. Алгоритм преобразования контекстно-свободной грамматики к грамматике в нормальной форме Хомского.
35. Определение идентификации.
36. Определяющее и использующее вхождение лексем.
37. Простейший способ реализации идентификации.
38. Ошибки, обнаруживаемые при идентификации.
39. Определение атрибутивной индукции. Пример.
40. Эквивалентные и корректные преобразования. Оптимизирующие преобразования.
41. Оптимизация на линейных участках: свёртка констант.
42. Оптимизация на линейных участках: удаление лишних операций.

Примеры задач:

1. Показать работу LR(1)-анализатора на примере разбора строки $!(a+b)*(b+c)!$
2. Показать работу метода рекурсивного спуска на примере разбора строки $!a+b*(b+a)!$
3. Показать работу метода простого предшествования на примере разбора строки $!a*b+b*a!$
4. Показать работу 1-предсказывающего нисходящего анализатора на примере разбора строки $!a*(b*(a+c)+b)!$
5. Показать работу метода Кока-Янгера-Касами на примере разбора строки $!(a+b)*(a+c)!$
6. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:
 $a:=x; b:=5; b:=b+1;$
 $c:=b-3; d:=b+c*(b+1);$
 $b:=z;$
 $z:=(x+d)*(b-3);$
7. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:
 $a:=0; c:=a-3;$
 $d:=a+c-1;$
 $b:=(a-3)*c+3*d;$
 $a:=x; r:=(d+1)/(a-3);$
8. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:
 $d:=2; c:=d-2;$
 $b:=d*d*c; x:=d*(b-3)*(d-2);$
 $d:=y; z:=(d-x)/(d-2);$
9. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:
 $b:=3; c:=5;$
 $a:=(b-c)*(c+b); d:=a+c;$
 $a:=x; b:=(c+b)/(a+c);$
10. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:
 $a:=10; c:=a*a; d:=a*(c-3);$
 $b:=d*d/2; a:=x;$
 $d:=(a*a)/(c-3);$

Результаты зачета с оценкой определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

«Отлично» - студент верно ответил на основные и дополнительные вопросы, показал работу одного из методов на предложенном примере

«Хорошо» - студент ответил на основные, но не все дополнительные вопросы, и/или допустил ошибки при решении задачи, и/или его ответ на основные вопросы содержал неточности.

«Удовлетворительно» - при ответе на основные и/или дополнительные вопросы допустил серьезные ошибки

«Неудовлетворительно» - студент не смог ответить на оба вопроса или на один вопрос и не решил задачу, не смог ответить на дополнительные вопросы

11. Учебно-методическое обеспечение

а) Электронный учебный курс по дисциплине в электронном университете «IDO» - <https://moodle.tsu.ru/course/view.php?id=28645>

б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине (Приложение 1).

12. Перечень учебной литературы и ресурсов сети Интернет

а) основная литература:

1. А. Ахо, Р.Сети, Дж. Ульман, Компиляторы: принципы, технологии, инструменты, Вильямс, 2003 г., 768 с.

2. Лебедев В.Н., Введение в системы программирования, Москва: Статистика, 1975 г., 312 с.

3. Грис, Д., Конструирование компиляторов для цифровых вычислительных машин, Мир, 1975 г., 544 с.

4. Ахо А., Ульман Дж., Теория синтаксического анализа, перевода и компиляции, Мир, 1978 г., 1104 с.

б) дополнительная литература:

5. Кнут Д., Искусство программирования, Мир, 1976 г., 736 с.

6. Пахомов Б.И., С/С++ и MS Visual C++ 2008 для начинающих, БХВ-Петербург, 2009 г., 642 с.

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:
MS Visual Studio C++, Borland C++ Builder

б) информационные справочные системы:

– Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>

– Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>

– ЭБС Лань – <http://e.lanbook.com/>

– ЭБС Консультант студента – <http://www.studentlibrary.ru/>

– Образовательная платформа Юрайт – <https://urait.ru/>

– ЭБС ZNANIUM.com – <https://znanium.com/>

– ЭБС IPRbooks – <http://www.iprbookshop.ru/>

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения занятий семинарского типа, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

15. Информация о разработчиках

Провкин Виктор Алексеевич, ассистент кафедры компьютерной безопасности