Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

Московский государственный университет имени М.В. Ломоносова

Механико-математический факультет

Кафедра Теоретической информатики

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.В.Михалев /

22 января 2019 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Наименование дисциплины:**

специальный курс по выбору кафедры

**Теория компиляции**

**Уровень высшего образования:**

**специалитет**

**Направление подготовки (специальность):**

**01.05.01 Фундаментальные математика и механика**

**Направленность (профиль) ОПОП:**

**Фундаментальные математика**

Форма обучения:

очная

Рабочая программа рассмотрена и одобрена

на заседании кафедры Теоретической информатики

(протокол № 2а от 22 января 2019 года)

Москва 2019

Рабочая программа дисциплины разработана в соответствии с самостоятельно установленным МГУ образовательным стандартом (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования по специальности «Фундаментальные математика и механика » (программы специалитета) в редакции приказа МГУ от 30 декабря 2016 г.

Год (годы) приема на обучение\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.** Место дисциплины в структуре ОПОП ВО. **Вариативная часть ОПОП ВО***.*

**2.** Входные требования для освоения дисциплины , предварительные условия : знание основ теории множеств, теории графов, алгебры и программирования на языках C и C++.

**3.** Результаты обучения по дисциплине , соотнесенные с требуемыми компетенциями выпускников*.*

|  |  |
| --- | --- |
| **Компетенции выпускников (коды)** | **Планируемые результаты обучения по дисциплине , соотнесенные с компетенциями** |
| *ПК-1, ПК-5* | ***Знать:*** *основы теории формальных языков и грамматик, классификацию языков, основные определения и теоремы теории формальных языков и конечных автоматов, алгоритмы и методы лексического и синтаксического разбора, методы генерации кода и оптимизации.*  ***Уметь*** *определять классы конкретных языков, выписывать детерминированные контекстно свободные грамматики языков и строить конечные автоматы для решения задач разбора и генерации, разрабатывать парсеры и лексические анализаторы, используя язык C/C++ и утилиты grep, lex, yacc.*  ***Владеть:*** *прикладными методами решения практических задач по изученным темам.* |
| *ОПК-1, ОПК-2, ОПК-4* | ***Уметь*** *решать стандартные задачи,**связанные с разбором формальных языков, построением конечных автоматов и регулярных выражений и генерацией кода.*  ***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***  ***Иметь опыт*** *использования на практике математических алгоритмов, в том числе с применением современных вычислительных систем.*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
| *УК-1* | ***Уметь*** *формулировать научно обоснованные гипотезы, создавать теоретические модели явлений и процессов, применять методологию научного познания в профессиональной деятельности.*  ***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***  ***Владеть*** *специальными разделами фундаментальной математики, методами анализа и решения задач.*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |

**4.** Формат обучения -- очный с использованем компьютера для демонстрации работы алгоритмови решения учебных задач*.*

**5.** Объем дисциплины составляет **7** з.е., в том числе **72**  академических часа, отведенных на контактную работу обучающихся с преподавателем, **38**  академических часов на самостоятельную работу обучающихся.

**6.** Содержание дисциплины , структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование и краткое содержание разделов и тем дисциплины ,**  **Форма промежуточной аттестации по дисциплине** | **Всего**  **(часы**) | В том числе | | | |
| **Контактная работа  (работа во взаимодействии с преподавателем)**  **Виды контактной работы, часы** | | | **Самостоятельная работа обучающегося,**  **часы** |
| Занятия лекционного типа\* | Занятия семинарского типа\* | **Всего** |  |
| Тема 1. Определения формальных языков и формальных грамматик. Иерархия Хомского. Грамматики без ограничения и контекстно зависимые грамматики. Примеры контекстно зависимых языков. | 6 | 4 |  | 4 | 2 |
| Тема 2. Контекстно свободные грамматики и языки. Левые и правые выводы, синтаксические деревья. Лемма о разрастании для контекстно свободных языков. Пример контекстно зависимого языка, не являющегося контекстно свободным. | 6 | 4 |  | 4 | 2 |
| Тема 3. Конечное автоматы и автоматные языки. Детерминированные и недетерминированные конечные автоматы, алгоритм построения эквивалентного детерминированного автомата по исходному недетерминированному. Праволинейные и леволинейные грамматики. Совпадения классов автоматных, леволинейных и праволинейных языков. Лемма о разрастании для автоматных языков. | 6 | 4 |  | 4 | 2 |
| Тема 4. Регулярные языки и выражения. Теорема Клини о совпадении классов регулярных и автоматных языков. Алгоритмы построения конечного автомата по регулярному выражению и, обратно, регулярного выражения по конечному автомату. | 6 | 4 |  | 4 | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тема 5. Минимальный детерминированный конечный автомат для данного автоматного языка. Изоморфизм всех минимальных детерминированных конечных автоматов, задающих данный язык. Алгоритм минимизации детерминированного конечного автомата. | 6 | | 4 |  | 4 | 2 |
| Тема 6. Операции над автоматными и контекстно свободными языками. Пример двух контекстно свободных языков, пересечение которых не является контекстно свободным языком. Пример контекстно свободного языка, дополнение к которому не является контекстно свободным. | 6 | | 4 |  | 4 | 2 |
| Текущий контроль успеваемости | 2 | |  | 2 | 2 |  |
| Тема 7. Преобразование контекстно свободных грамматик. Удаление эпсилон правил и цепных правил. Нормальная форма Хомского контекстно свободной грамматики. | 6 | | 4 |  | 4 | 2 |
| Тема 8. Задача разбора. Грамматики с маркером конца строки. Рекурсивный, или нисходящий, разбор, восстанавливающий левые выводы цепочек языка. Класс контекстно свободных грамматик, допускающих рекурсивный разбор: LL(1)-грамматики (определение и примеры). Преобразование грамматики к виду LL(1) (избавление от левой рекурсии) в тех случаях, когда это возможно, примеры. Программная реализация LL(1)-парсера, примеры. | 6 | | 4 |  | 4 | 2 |
| Тема 9. Восходящий разбор, или разбор с помощью конечного автомата со стеком. Определение LR-процесса и возможных действий (сдвиг и свертки по правилам грамматики). Соответствие между успешными LR-процессами и правыми выводами цепочек языка. Возможные неоднозначности при построении LR-процесса. | 6 | | 4 |  | 4 | 2 |
| Тема 10. Разрешение неоднозначности LR-процесса путем построения множества состояний LR-разбора. Определение LR(0)-ситуации разбора. Определение LR(0)-состояния разбора как конечного множества LR(0)-ситуаций. Алгоритм построения множества состояний LR(0)-разбора. Построение таблицы действий (сдвиг и свертка по правилу) для LR(0)-разбора, примеры. Возможные конфликты типа сдвиг-свертка или свертка-свертка. Определение LR(0)-грамматики через отсутствие конфликтов в построенном множестве. Примеры LR(0) и не LR(0)-грамматик. Алгоритм работы LR(0)-парсера. | 6 | | 4 |  | 4 | 2 |
| Тема 11. LR(1)-разбор. Определение LR(1)-ситуации разбора и LR(1)-состояния разбора как конечного множества LR(1)-ситуаций. Алгоритм построения множества состояний LR(1)-разбора. Построение таблицы действий (сдвиг и свертка по правилу) для LR(1)-разбора, примеры. Возможные конфликты типа сдвиг-свертка или свертка-свертка. Определение LR(1)-грамматики через отсутствие конфликтов в построенном множестве. Примеры LR(1) и не LR(1)-грамматик. Алгоритм работы LR(1)-парсера. Соответствие стека состояний LR(1) разбора и стека LR-процесса. | 6 | | 4 |  | 4 | 2 |
| Тема 12. Использование семантического стека параллельно стеку состояний LR(0) или LR(1) разбора для решения задачи компиляции, т.е. перевода с одного языка на другой. Примеры. | 6 | | 4 |  | 4 | 2 |
| Текущий контроль успеваемости – контрольная рмбота | 2 | |  | 2 | 2 |  |
| Тема 13. Программные средства для построения компиляторов: LEX (или FLEX) и YACC (или BISON). Построение синтаксического анализатора, или парсера, с помощью системы YACC (BISON). Входной язык описания парсера. Использование сканера, реализованного с помощью LEX, как поставщика информации для парсера. Описание грамматики языка средствами YACC. Способы разрешения конфликтов с помощью приписывания приоритета лексемам и правилам. Семантический стек парсера и определение семантических действий во входном языке для YACC. Примеры построения простых компиляторов с помощью YACC. | 6 | | 4 |  | 4 | 2 |
| Тема 14. Модельный компилятор, реализованный с помощью утилит LEX и YACC. Описание языка. Сканер языка и первый этап реализации парсера: разбор синтаксиса без генерации кода (syntax checker). | 6 | | 4 |  | 4 | 2 |
| Тема 15. Разбор выражений: генерация дерева выражения в результате синтаксического разбора. | 6 | | 4 |  | 4 | 2 |
| Тема 16. Таблицы символов, их структура и использование при написании компилятора. Способы представления промежуточного кода: язык стековой машины, язык RTL. Алгоритм генерации кода для логических выражений. | 6 | | 4 |  | 4 | 2 |
| Тема 17. Генерация ассемблерного кода по RTL. Распределение физических регистров и генерация кода для команд с ограничениями на использование регистров. | 6 | | 4 |  | 4 | 2 |
| Тема 18. Алгоритмы оптимизации кода в случае RTL-представления промежуточного кода. Оптимизация на уровне базового блока: удаление мертвых выражений, нахождение общих подвыражений. Глобальная оптимизация: вычисление инвариантов циклов до начала цикла, strength reduce optimization. | 6 | | 4 |  | 4 | 2 |
| Консультации | 4 | | | | 4 |  |
| Промежуточная аттестация – контрольная работа | 2 | | | |  | 2 |
| Экзамен | 4 | | | | 4 |  |
| **Итого** | 110 | 72 | | | | 38 |

7. Фонд оценочных средств (ФОС) для оценивания результатов обучения по дисциплине

7.1. Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.

1. Выписать контекстно зависимую грамматику для языка L = { *anbncn*, *n*=1, 2, … }

2. Выписать однозначную контекстно свободную грамматику для языка всех слов в алфавите {*a*, *b*}, имеющих одинаковую степень по *a* и *b*.

3. Применив лемму о разрастании, показать, что конкретный язык (например, L = {u u, где u – любая цепочка из букв *a*, *b*}) не является контекстно свободным.

4. Для конкретных недетерминированных конечных автоматов построить эквивалентные им детерминированные автоматы.

5.Для конкретных конечных автоматов выписать регулярное выражение, задающее тот же самый язык (выражение строится по алгоритму, который используется в доказательстве теоремы Клини).

6. Для конкретной грамматики с маркером конца цепочки построить систему состояний LR(0)-разбора и проверить, есть ли в ней конфликты (т.е. допускает ли она LR(0)-разбор).

7. Привести пример однозначной контекстно свободной грамматики, не допускающей LR(1)-разбор.

7.2. Типовые контрольные задания или иные материалы для проведения промежуточной аттестации.

1. Для конкретный языков определить, к какому классу в иерархии Хомского они принадлежат.

2. Для конкретного языка построить однозначную контекстно свободную грамматику.

3. По заданному недетерминированному конечному автомату построить эквивалентный ему детерминированный конечный автомат.

4. По заданному конечному автомату построить регулярное выражение, задающее тот же самый язык.

5. По заданному детерминированному конечному автомату построить эквивалентный ему минимальный детерминированный конечный автомат.

6. Привести конкретную контекстно свободную грамматику к нормальной форме Хомского.

7. Определить, допускает ли конкретная контекстно свободная грамматика рекурсивный разбор.

8. Определить, допускает ли конкретная контекстно свободная грамматика LR(0)-разбор.

9. Определить, допускает ли конкретная контекстно свободная грамматика LR(1)-разбор.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине (модулю)** | | | | |
| Оценка  РО и соответствующие виды оценочных средств | 2 | 3 | 4 | 5 |
| **Знания**  *(виды оценочных средств: устные и письменные опросы и контрольные работы, тесты, и т.п. )* | Отсутствие знаний | Фрагментарные знания | Общие, но не структурированные знания | Сформированные систематические знания |
| **Умения**  *(виды оценочных средств: практические контрольные задания, написание и защита рефератов на заданную тему и т.п.)* | Отсутствие умений | В целом успешное, но не систематическое умение | В целом успешное, но содержащее отдельные пробелы умение (допускает неточности непринципиального характера) | Успешное и систематическое умение |
| **Навыки  (владения, опыт деятельности)**  *(виды оценочных средств: выполнение и защита курсовой работы, отчет по практике, отчет по НИР и т.п.)* | Отсутствие навыков (владений, опыта) | Наличие отдельных навыков (наличие фрагментарного опыта) | В целом, сформированные навыки (владения), но используемые не в активной форме | Сформированные навыки (владения), применяемые при решении задач |

8. Ресурсное обеспечение:

* Перечень основной и дополнительной литературы,
* **А. Основная литература**
* 1. *Пентус А.Е., Пентус М.Р.* Математическая теория формальных языков. — Интернет-университет информационных технологий www.intuit.ru. — Москва, "Бином", 2006. — 247 стр.
* 2. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ. — Москва, Мир, 1978. — 612 стр.
* **Б. Дополнительная литература**
* 1. *Ахо А., Сети Р., Ульман Дж.* Компиляторы: принципы, технологии и инструменты. — Москва, Вильямс, 2001. — 768 стр.

9. Язык преподавания.

Русский

10. Преподаватель

Борисенко В.В.

11. Автор программы.

Борисенко В.В.