

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«24» июня 2025 г.
Протокол №2

**Рабочая программа дисциплины (модуля)
«Теория языков программирования»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Искусственный интеллект

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2025

**Москва
2025**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	4
3. Тематический план	6
4. Содержание дисциплины (модуля)	6
5. Учебно-методическое обеспечение	7
6. Материально-техническое обеспечение	7
7. Методические и оценочные материалы	9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Теория языков программирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Искусственный интеллект, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Теория языков программирования» позволяет глубже понять структуру, синтаксис и семантику языков, что способствует эффективному проектированию и анализу программных систем. Это знание важно для создания новых языков, компиляторов и инструментов разработки, обеспечивая повышение качества и надёжности программного обеспечения.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Искусственный интеллект и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 2, 3 или 4 курсе в 4, 5, 6, 7, 8 семестрах на выбор.

Цель изучения дисциплины (модуля): формирование фундаментальных знаний о структуре, синтаксисе и семантике языков программирования для эффективного проектирования, анализа и реализации программных систем.

Задачи изучения дисциплины (модуля):

- изучить основные концепции теории языков программирования, включая формальные грамматики, автоматы и семантику;
- освоить методы анализа и классификации языков программирования по парадигмам и типам;
- научиться проектировать и реализовывать простые языки программирования с использованием теоретических основ.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- классификации формальных языков и грамматик;
- основные алгоритмы разбора для контекстно-свободных грамматик;
- важнейшие подклассы КС-грамматик;
- практическое применение автоматов и грамматик;

уметь:

- определять класс грамматик, необходимый для решения задачи;
- строить конечный автомат (регулярное выражение) по описанию автоматного языка;
- строить автомат с магазинной памятью по описанию контекстно-свободного языка;

владеть:

- математическим аппаратом теории формальных языков.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализации математических моделей в профессиональной деятельности
ОПК-6	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области искусственного интеллекта
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных

			технологий программирования в области искусственного интеллекта
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований в области искусственного интеллекта
ПК-1.	Способен формулировать задачи с математической точностью, обосновывать утверждения строго и анализировать полученные результаты в области математики и компьютерных наук	ПК-1.1.	Обладает базовыми знаниями, полученными в области математических наук, программирования и информационных технологий
		ПК-1.2.	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в математике и информатике
		ПК-1.3.	Имеет опыт работы с задачами в области математики и компьютерных наук, включая применение математических методов для решения практических задач

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Введение	4	4	2	17	Домашние задания, Контрольная работа
2	Регулярные выражения	4	4	2	17	Контрольная работа
3	Конечные автоматы	4	4	2	17	Контрольная работа
4	Контекстно-свободные грамматики	6	6	2	18	Домашние задания, Контрольная работа
5	Алгоритмы синтаксического анализа	4	4		17	Домашние задания, Коллоквиум
6	Перевод и преобразования	4	4		17	Домашние задания, Коллоквиум
7	Семантический анализ	4	4	2	17	Контроль теоретических знаний Проект
	<i>Зачет с оценкой</i>					
	<i>Итого:</i>	<i>30</i>	<i>30</i>	<i>10</i>	<i>120</i>	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	<i>190</i>				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	<i>5</i>				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Введение	Обзор формальных языков и иерархии Хомского
2	Регулярные выражения	Регулярные выражения и регулярные языки
3	Конечные автоматы	Конечные автоматы: DFA и NFA. Преобразования DFA ↔ NFA, pumping-лемма, свойства замыкания
4	Контекстно-свободные грамматики	Контекстно-свободные грамматики: основные понятия и нормальные формы. Автоматы с магазинной памятью и связь с КС-языками. Деревья синтаксического анализа и неоднозначность грамматик
5	Алгоритмы синтаксического анализа	LL(1)-парсинг, рекурсивный спуск и предиктивные таблицы. LR(0) и SLR(1)-парсеры. Канонический LR(1)-парсинг и генерация таблиц. LALR(1)-парсеры и инструменты генерации
6	Перевод и преобразования	Конечные преобразователи и регулярные отношения
7	Семантический анализ	Атрибутные грамматики, синтаксически-направленный перевод и семантические действия. Типовые системы, статический анализ и проверки семантики. Выполнение итогового проекта

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Малявко, А. А. Формальные языки и компиляторы : учебник для вузов / А. А. Малявко. — Москва : Издательство Юрайт, 2025. — 429 с. — (Высшее образование). — ISBN 978-5-534-04288-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562822>.

2. Довек, Ж. Введение в теорию языков программирования : научное издание / Ж. Довек, Ж.-Ж. Леви ; пер. с англ. В. Н. Брагилевского, А. М. Пеленицына. - 2-е изд. - Москва : ДМК Пресс, 2023. - 135 с. - ISBN 978-5-89818-582-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2107945>.

3. Орлов, С. А. Теория и практика языков программирования : учебник для вузов / С. А. Орлов. - Санкт-Петербург : Питер, 2021. - 688 с. - (Стандарт 3-го поколения). - ISBN 978-5-4461-9740-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1857045>.

4. Тюгашев, А. А. Языки программирования : учебное пособие. Стандарт третьего поколения / А. А. Тюгашев. - Санкт-Петербург : Питер, 2021. - 334 с. - ISBN 978-5-4461-9407-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2143578>.

5. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561744>.

6. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 196 с. — (Высшее образование). — ISBN 978-5-534-18759-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560807>.

Дополнительная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/581329>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и

индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		

Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Теория языков программирования» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, контрольные работы, контроль теоретических знаний, коллоквиум, домашние задания, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Участие в семинаре (аудиторная работа) – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их.

При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Коллоквиум – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки к коллоквиуму необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Контроль теоретических знаний – устные ответы на вопросы, список которых известен студенту заранее.

В процессе подготовки студенту необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Теория языков программирования»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине. Для оценивания текущего контроля успеваемости и промежуточной

аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	Зачтено	
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает
4	Удовлетворительно	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Теория языков программирования» оценивается следующим образом:

Активность	Вес	Описание
Контрольные работы	40%	Письменная работа с набором задач, которые нужно решить за ограниченное время
Контроль теоретических знаний	30%	Ответы на вопросы, список которых известен студенту заранее по пройденным темам дисциплины (модуля)
Коллоквиум	30%	Устные ответы на вопросы, список которых известен студенту заранее

Формула расчёта итоговой оценки по дисциплине (модулю) «Теория языков программирования»: $\langle 0,4 \times \text{среднее за контрольные работы} + 0,3 \times \text{контроль теоретических знаний} + 0,3 \times \text{среднее за коллоквиум} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1

1. Опишите иерархию Хомского, перечислив типы грамматик и соответствующие им языки с примерами для каждого типа.
2. Постройте регулярное выражение для языка, состоящего из строк, начинающихся с "ab" и заканчивающихся "ba", где между ними может быть любое количество символов "a" или "b".
3. Преобразуйте регулярное выражение $(a|b)^*ab$ в недетерминированный конечный автомат (NFA), нарисуйте его диаграмму и объясните работу.
4. Докажите с помощью pumping-леммы, что язык $\{a^n b^n \mid n \geq 0\}$ не является регулярным.
5. Постройте детерминированный конечный автомат (DFA) для языка, состоящего из строк над алфавитом $\{0, 1\}$, где количество единиц четное, и проверьте его работу на примерах строк "010", "111" и "000".

Домашнее задание 2

6. Постройте контекстно-свободную грамматику (КСГ) для языка арифметических выражений с операторами +, * и скобками, и приведите пример дерева синтаксического анализа для строки "a + (b * c)".

7. Приведите КСГ в нормальную форму Хомского и объясните, почему это полезно для анализа.

8. Разработайте предиктивную таблицу для LL(1)-парсера на основе простой КСГ (например, для выражений с + и *) и продемонстрируйте разбор строки "a + b * c".

9. Постройте LR(0)-автомат для грамматики $S \rightarrow aSb \mid \epsilon$ и объясните, является ли грамматика SLR(1).

10. Используя инструмент (например, yacc или аналог), сгенерируйте LALR(1)-парсер для КСГ арифметических выражений и протестируйте его на корректной и некорректной строках.

Домашнее задание 3

11. Постройте конечный преобразователь для регулярного отношения, которое заменяет каждую подстроку "ab" на "ba" в строке над алфавитом {a, b}.

12. Опишите, как регулярные отношения связаны с конечными преобразователями, и приведите пример преобразования строки "abab" в "baba".

13. Разработайте атрибутивную грамматику для вычисления значения простого арифметического выражения (например, с + и числами), и продемонстрируйте синтаксически-направленный перевод для строки "2 + 3 * 4".

14. Выполните статический анализ семантики для программы на простом языке (например, проверка типов в присваиваниях), используя атрибутивные правила, и укажите возможные ошибки.

15. В рамках итогового проекта реализуйте простой интерпретатор для мини-языка (например, с переменными и циклами), используя атрибутивную грамматику для семантического анализа, и протестируйте его на примере программы.

Примерные задания для контроля теоретических знаний

Введение (Обзор формальных языков и иерархии Хомского) (3 задания)

1. **MCQ:** Какая грамматика генерирует рекурсивно-перечислимые языки?
 - a) Тип 0
 - b) Тип 1
 - c) Тип 2
 - d) Тип 3
2. **Короткий ответ:** Перечислите четыре типа грамматик в иерархии Хомского и укажите, какие языки они генерируют.
3. **True/False:** Все регулярные языки являются контекстно-свободными. Объясните свой ответ.

Регулярные выражения (Регулярные выражения и регулярные языки) (4 задания)

4. **Задача:** Постройте регулярное выражение для языка всех строк над алфавитом {0, 1}, содержащих подстроку "01" хотя бы один раз.
5. **MCQ:** Какое регулярное выражение эквивалентно языку всех строк, начинающихся и заканчивающихся на 'a'?
 - a) $a(a|b)^*a$
 - b) $(a|b)a$
 - c) $a(a/b)$
 - d) $(a|b)+a$
6. **Короткий ответ:** Используя pumping-лемму, докажите, что язык $\{a^n b^n \mid n \geq 0\}$ не является регулярным.

7. **Задача:** Упростите регулярное выражение $(a|b)^*(aa|b)$ и объясните шаги.

Конечные автоматы (DFA и NFA, преобразования, pumping-лемма, свойства замыкания) (4 задания)

8. **Задача:** Постройте DFA для языка строк над $\{a, b\}$, где число 'a' делится на 3.
9. **MCQ:** Что такое ϵ -замыкание в NFA?
- a) Множество состояний, достижимых без чтения символов
 - b) Множество всех возможных переходов
 - c) Начальное состояние автомата
 - d) Принимающее состояние
10. **Короткий ответ:** Объясните, как преобразовать NFA в DFA, и приведите пример для регулярного выражения a^* .
11. **True/False:** Регулярные языки замкнуты относительно дополнения. Объясните, почему.

Контекстно-свободные грамматики (Основные понятия, нормальные формы, PDA, деревья, неоднозначность) (4 задания)

12. **Задача:** Постройте КСГ для языка $\{a^i b^j c^k \mid i = j \text{ или } j = k\}$, и приведите дерево разбора для строки "abbc".
13. **MCQ:** В нормальной форме Хомского правила имеют вид:
- a) $A \rightarrow BC$ или $A \rightarrow a$
 - b) $A \rightarrow aB$ или $A \rightarrow \epsilon$
 - c) $S \rightarrow \alpha$, где α — строка
 - d) Любые правила
14. **Короткий ответ:** Опишите, как PDA распознает язык $\{a^n b^n c^n\}$, и укажите ключевые компоненты.
15. **Задача:** Приведите пример неоднозначной КСГ и два разных дерева разбора для одной строки.

Алгоритмы синтаксического анализа (LL(1), рекурсивный спуск, LR(0), SLR(1), LALR(1), инструменты) (5 заданий)

16. **Задача:** Постройте FIRST и FOLLOW множества для грамматики $S \rightarrow AB, A \rightarrow a \mid \epsilon, B \rightarrow b$.
17. **MCQ:** Какой парсер использует таблицу предиктивного анализа?
- a) LR(0)
 - b) LL(1)
 - c) SLR(1)
 - d) LALR(1)
18. **Короткий ответ:** Объясните разницу между SLR(1) и LALR(1), и приведите пример, где LALR(1) имеет преимущество.
19. **Задача:** Реализуйте рекурсивный спуск для простого арифметического выражения с операторами + и * (без приоритетов).
20. **True/False:** LR(0)-парсер может иметь сдвиг-редукционные конфликты. Объясните.

Перевод и преобразования (Конечные преобразователи и регулярные отношения) (3 задания)

21. **Задача:** Постройте конечный преобразователь, который заменяет каждую 'a' на 'ab' в строке.
22. **MCQ:** Регулярное отношение — это:
- a) Подмножество A^*
 - b) Подмножество $A^* \times B^*$

- c) Конечный автомат
 - d) Грамматика
23. **Короткий ответ:** Опишите, как конечный преобразователь реализует перевод "a" → "01", "b" → "10".

Семантический анализ (Атрибутные грамматики, синтаксически-направленный перевод, типовые системы, проект) (3 задания)

24. **Задача:** Разработайте атрибутную грамматику для проверки, что все переменные в выражении объявлены (простой пример с присваиваниями).
25. **МСQ:** В синтаксически-направленном переводе атрибуты вычисляются:
- a) Только сверху вниз
 - b) Только снизу вверх
 - c) В любом порядке
 - d) Только в листьях дерева

Примерный перечень вопросов для коллоквиума

1. Что такое регулярное выражение и как оно описывает регулярные множества?
2. Какие основные операции используются в регулярных выражениях?
3. Как регулярные выражения применяются для поиска шаблонов в тексте?
4. В чём заключается эквивалентность регулярных выражений и конечных автоматов?
5. Как регулярные выражения используются в лексическом анализе компиляторов?
6. Какие ограничения накладывают регулярные выражения на описание вложенных структур?
7. Что представляет собой модель конечного автомата с конечным числом состояний?
8. В чём разница между детерминированным (ДКА) и недетерминированным конечным автоматом (НКА)?
9. Как доказать эквивалентность ДКА и НКА?
10. Каким образом конечные автоматы распознают регулярные языки?
11. Как конечные автоматы применяются в лексическом анализе?
12. Опишите процесс минимизации состояний конечного автомата.
13. Какие преимущества даёт минимизация конечного автомата?
14. Что такое контекстно-свободная грамматика и как она описывает синтаксис языков?
15. Какова форма правил продукции в КС-грамматиках?
16. Почему КС-грамматики поддерживают описание вложенных конструкций?
17. Как КС-грамматики используются как основа синтаксического анализа?
18. Какие ограничения существуют у контекстно-свободных грамматик с точки зрения выразительности?
19. Как строится дерево разбора на основе грамматики и входной строки?
20. В чём суть нисходящих методов синтаксического анализа?
21. Как работают восходящие методы синтаксического анализа?
22. Что такое LR-анализаторы и в чём их особенность?
23. Почему детерминизм важен в LR-анализаторах?
24. Какие методы используются для обработки и восстановления ошибок в синтаксическом анализе?
25. Как осуществляется генерация парсеров на основе грамматик?
26. Что такое конечный преобразователь и как он отличается от конечного автомата?
27. Как конечные преобразователи используются для преобразования лексем в токены?
28. В чём отличие детерминированных и недетерминированных конечных преобразователей?

29. Как конечные преобразователи применяются в трансляции и оптимизации кода?
30. Как формально описываются преобразования, выполняемые конечными преобразователями?

Примерные задания по контрольным работам

Контрольная работа по теме «Конечные автоматы»

1. Постройте детерминированный конечный автомат (ДКА), распознающий язык всех строк над $\{0,1\}$, содержащих чётное число нулей.
2. Составьте недетерминированный конечный автомат (НКА) для языка всех строк, содержащих подстроку «101».
3. Преобразуйте заданный НКА в эквивалентный ДКА.
4. Докажите эквивалентность двух конечных автоматов, приведя соответствие их состояний и переходов.
5. Распознайте, используя построенный ДКА, принадлежит ли строка «010101» заданному языку.
6. Минимизируйте данный ДКА и объясните шаги минимизации.
7. Опишите, как конечные автоматы применяются в лексическом анализе компилятора.
8. Разработайте конечный автомат, распознающий все строки над $\{a,b\}$, в которых количество a кратно трём.
9. Объясните, почему регулярные языки эквивалентны языкам, распознаваемым конечными автоматами.
10. Сравните преимущества и недостатки ДКА и НКА в контексте реализации лексического анализатора.

Контрольная работа по теме «Алгоритмы синтаксического анализа»

1. Запишите контекстно-свободную грамматику для арифметических выражений с операциями сложения и умножения.
2. Постройте дерево разбора для выражения « $a + b * c$ » по заданной грамматике.
3. Опишите нисходящий метод синтаксического анализа и приведите пример его применения.
4. Опишите восходящий метод синтаксического анализа и сравните его с нисходящим.
5. Постройте LR(0)-автомат для заданной грамматики.
6. Объясните понятие детерминизма в LR-анализаторах и его значение.
7. Реализуйте обработку ошибок при синтаксическом анализе с помощью метода паники.
8. Опишите стратегию восстановления ошибок синтаксического анализа с сохранением максимально возможного контекста.
9. Составьте пример кода генерации парсера на основе таблиц LR-анализатора.
10. Проанализируйте преимущества использования автоматических генераторов парсеров в разработке компиляторов.

Примерное описание проекта

Название проекта: Реализация парсера для простого арифметического языка с использованием LL(1)- и LR(1)-алгоритмов

Цель проекта:

Целью проекта является практическое освоение алгоритмов синтаксического анализа, включая LL(1)-парсинг с предиктивными таблицами, рекурсивный спуск, а также LR(0), SLR(1) и LALR(1)-парсеры. Студенты должны продемонстрировать умение реализовать парсер для заданного контекстно-свободного языка, провести анализ его свойств (например, LL(1)-свойства, наличие конфликтов в LR-таблицах) и сравнить эффективность

различных подходов. Проект способствует глубокому пониманию генерации таблиц анализа, обработки ошибок и интеграции с инструментами типа yacc или ANTLR.

Задачи проекта:

1. Определить контекстно-свободную грамматику для простого языка (например, арифметические выражения с операторами +, -, *, /, скобками и числами).
2. Провести анализ грамматики: вычислить FIRST и FOLLOW множества, проверить LL(1)-свойства и построить предиктивную таблицу.
3. Реализовать LL(1)-парсер с использованием рекурсивного спуска или таблиц.
4. Реализовать LR(1)-парсер (SLR(1) или LALR(1)), включая генерацию таблиц действий и переходов.
5. Добавить обработку синтаксических ошибок и сравнить производительность парсеров на тестовых примерах.
6. Интегрировать проект с инструментом генерации парсеров (например, написать код на Python или C++ с использованием библиотек).
7. Подготовить отчет с анализом неоднозначностей, конфликтов и рекомендациями по оптимизации.

Этапы выполнения проекта:

1. **Подготовительный этап (1-2 недели):** Изучить теоретический материал по LL(1) и LR-парсерам. Выбрать и формализовать грамматику языка. Вычислить FIRST/FOLLOW и построить таблицы вручную.
2. **Реализация LL(1)-парсера (2-3 недели):** Написать код для предиктивного парсера с рекурсивным спуском. Протестировать на корректных и некорректных строках.
3. **Реализация LR-парсера (2-3 недели):** Построить автоматы для LR(0), SLR(1) и LALR(1). Реализовать генерацию таблиц и парсер. Провести тестирование и сравнение с LL(1).
4. **Интеграция и тестирование (1 неделя):** Объединить компоненты, добавить обработку ошибок и использовать инструмент генерации (например, ANTLR для прототипа).
5. **Анализ и отчет (1 неделя):** Составить отчет с примерами, диаграммами и выводами. Подготовить презентацию для защиты.

Критерии защиты проекта:

- Проект должен быть представлен в виде работающего программного кода (на Python, Java или C++), отчет в формате PDF и презентация (5-10 слайдов).
- Защита включает демонстрацию работы парсера на примерах (корректные выражения, ошибки), объяснение алгоритмов и анализ результатов.
- Студент должен ответить на вопросы по теории (например, почему грамматика LL(1), как разрешаются конфликты в LR).
- Проект должен быть оригинальным; использование готовых библиотек допускается только для сравнения, но основная реализация — ручная.

Критерии оценивания проекта:

Оценка выставляется по шкале 0-10 баллов, с распределением:

- **Теоретическая часть (2 балла):** Корректность анализа грамматики (FIRST/FOLLOW, таблицы).

- **Реализация LL(1) (2 балла):** Правильность кода, обработка ошибок, эффективность.
- **Реализация LR (2 балла):** Построение таблиц, разрешение конфликтов, сравнение методов.
- **Тестирование и интеграция (1,5 балла):** Полнота тестов, использование инструментов, демонстрация работы.
- **Отчет и защита (1,5 балла):** Качество отчета (структура, примеры), презентация, ответы на вопросы.
- **Общие критерии (1 балл):** Оригинальность, соблюдение сроков, глубина анализа. Порог для зачета: 6 баллов. Максимальная оценка за инновационные решения (например, оптимизация для неоднозначных грамматик).

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите тип грамматики по Хомскому, который генерирует регулярные языки.	3	УК-1
2	Как называется лемма, используемая для доказательства нерегулярности языков?	pumping-лемма	УК-1
3	Укажите нормальную форму для контекстно-свободных грамматик, где все правила имеют вид $A \rightarrow BC$ или $A \rightarrow a$.	Хомского	УК-1
4	Какой парсер использует предиктивные таблицы для анализа?	LL(1)	УК-1
5	Что такое ϵ -замыкание в конечных автоматах?	множество состояний по ϵ -переходам	УК-1
6	Сколько типов грамматик в иерархии Хомского?	4	ОПК-1
7	Укажите класс языков, генерируемый типом 2 грамматик.	контекстно-свободные	ОПК-1
8	Какой автомат моделирует контекстно-свободные языки?	PDA	ОПК-1
9	Укажите метод анализа, использующий FIRST и FOLLOW множества.	LL(1)	ОПК-1
10	Что такое неоднозначность грамматики?	несколько деревьев разбора	ОПК-1
11	Какой алгоритм парсинга использует стек и таблицу действий?	LR	ОПК-6
12	Укажите тип LR-парсера, который минимизирует количество состояний.	LALR(1)	ОПК-6
13	Как называется инструмент для генерации парсеров?	уасс	ОПК-6
14	Что такое атрибутивная грамматика?	грамматика с атрибутами	ОПК-6
15	Укажите метод перевода, направляемый синтаксисом.	синтаксически-направленный перевод	ОПК-6
16	Укажите количество уровней в иерархии Хомского.	4	ПК-1
17	Какой тип грамматики соответствует типу 0?	неограниченные	ПК-1
18	Назовите одну операцию замыкания для регулярных языков.	объединение	ПК-1
19	Какой автомат имеет детерминированный переход?	DFA	ПК-1
20	Укажите нормальную форму, где правила начинаются с терминала.	Грейбаха	ПК-1