

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)

УТВЕРЖДАЮ
Заведующий кафедрой УИ
_____ А.Ф. Уваров
« ____ » _____ 2012 г.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ
по дисциплине**

«Методы трансляции»

Составлены кафедрой «Управление инновациями»
Для студентов, обучающихся по направлению подготовки 220600 «Инноватика»
Магистерская программа «Мультимедийные многопроцессорные системы на кристалле»

Форма обучения очная

Разработчик:

профессор, д.т.н.

_____ Ю.Л. Костюк
« ____ » _____ 2012 г.

Томск 2012 г.

Содержание

Введение.....	3
Практическое занятие № 1 Разработка автоматной грамматики, порождающей типичные лексемы языка программирования.	4
Практическое занятие № 2 Разработка и реализация конечного автомата по заданной автоматной грамматике.....	5
Практическое занятие № 3 Разработка и реализация лексического анализатора для выделения типичных лексем языка программирования.	6
Практическое занятие № 4 Разработка контекстно-свободной грамматики, порождающей типичные конструкции языка программирования.	7
Практическое занятие № 5 Преобразование контекстно-свободной грамматики к обобщенной нормальной форме Грейбах.	8
Практическое занятие № 6 Построение и реализация LL(1)-анализатора.	9
Практическое занятие № 7 Построение семантической таблицы в LL(1)-анализаторе для генерации обратной польской строки.	10
Практическое занятие № 8 Построение и реализация генератора обратной польской строки в виде LL(1)-анализатора.	11
Практическое занятие № 9 Построение и реализация интерпретатора обратной польской строки.....	12
Библиографический список.....	13

Введение

Изучение дисциплины «**Методы трансляции**» (уровень дисциплины региональный) имеет существенное значение в специальной подготовке студентов по направлению «Инноватика».

Цель данного пособия состоит в выработке навыков в разработке программных средств трансляции и интерпретации таких формальных языков, как языки программирования, языки запросов к информационным и интеллектуальным системам и др.

Для полноценного понимания и усвоения материала необходимо предварительно изучить дисциплины "Дискретная математика" и "Основы программирования".

Для углубленного изучения и освоения материала целесообразно применение различных форм самопроверки знаний студентов: тесты, задачи, упражнения. Они могут быть использованы при проведении практических занятий в университете, выполнении курсовых, контрольных и аудиторных работ, а также при самостоятельном изучении данных дисциплин.

Одним из наиболее интенсивных способов изучения дисциплины является самостоятельная реализация алгоритмов трансляции, изучаемых на лекциях. При этом вырабатывается опыт и навыки, необходимые при разработке сложных программных средств.

Предлагаемые задания позволят глубже освоить теоретические и практические вопросы теории формальных языков и методов трансляции, понять принципы описания и анализа языков программирования, научиться грамотно применять эти теоретические знания при разработке программных средств трансляции.

Практическое занятие № 1

Разработка автоматной грамматики, порождающей типичные лексемы языка программирования.

Цель занятия:

Научиться составлять и анализировать порождающие правила автоматной грамматики, порождающей типичные лексемы языка программирования.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов составления порождающих правил автоматной грамматики.

1.2. Освоение навыков анализа множества цепочек, порождаемых автоматной грамматикой.

1.3. Изучение методов составления порождающих правил для типичных лексем языка программирования.

Задание:

Разработать и дать анализ автоматной грамматики, порождающей типичные лексемы языка программирования, такие, как имена, изображения целых и вещественных чисел, символьные строки, составные символы (‘:=’ ‘<=’ и др.), отдельные символы (‘>’ ‘<’ ‘(’ ‘)’ и др.), комментарии, пробелы и др. разделители.

Методика выполнения:

Теория описана в [1], разд. 1. При составлении порождающих правил автоматной грамматики необходимо учитывать, чтобы получившаяся грамматика была детерминированной. При необходимости следует применить преобразования, описанные в [1], разд. 3.1, 3.2, 3.3 и 3.4.

Практическое занятие № 2

Разработка и реализация конечного автомата по заданной автоматной грамматике.

Цель занятия:

Научиться составлять и реализовывать конечный автомат для распознавания множества цепочек символов, порождаемых автоматной грамматикой.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов построения конечных автоматов на основе автоматной грамматики.

1.2. Освоение методов реализации конечных автоматов в виде алгоритмов распознавания.

1.3. Освоение методов тестирования конечного автомата.

Задание:

Разработать, реализовать в виде программы и протестировать конечный автомат по автоматной грамматике, разработанной на 1-м занятии.

Методика выполнения:

Теория описана в [1], разд. 3.1., 3.2. При построении конечного автомата по порождающим правилам автоматной грамматики необходимо учитывать, чтобы получившийся конечный автомат был детерминированным. При необходимости следует применить преобразования, описанные в [1], разд. 3.3, 3.4 и 3.5.

Практическое занятие № 3

Разработка и реализация лексического анализатора для выделения типичных лексем языка программирования.

Цель занятия:

Научиться разрабатывать и реализовывать лексический анализатор, распознающий и выделяющий из входного текста лексемы, порождаемые автоматной грамматикой.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов построения лексического анализатора на основе автоматной грамматики и конечного автомата.

1.2. Освоение методов реализации лексического анализатора в виде алгоритма распознавания, вызывающего семантические программы.

1.3. Изучение методов реализации семантических программ.

1.4. Освоение методов тестирования лексического анализатора.

Задание:

Разработать, реализовать в виде процедуры и протестировать лексический анализатор на основе конечного автомата, разработанного на 2-м занятии. Кроме перечисленных в 1-м задании лексем, лексический анализатор должен также распознавать служебные слова языка программирования, задаваемых в виде таблицы. Для тестирования написать вспомогательную тестирующую программу.

Методика выполнения:

Теория описана в [1], разд. 2 и 3.6.

Практическое занятие № 4

Разработка контекстно-свободной грамматики, порождающей типичные конструкции языка программирования.

Цель занятия:

Научиться составлять и анализировать порождающие правила контекстно-свободной грамматики, порождающей типичные конструкции языка программирования.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов составления порождающих правил контекстно-свободной грамматики.

1.2. Освоение навыков анализа множества цепочек, порождаемых контекстно-свободной грамматикой.

1.3. Изучение методов составления порождающих правил для типичных конструкций языка программирования.

Задание:

Разработать и дать анализ контекстно-свободной грамматики, порождающей такие типичные конструкции языка программирования, как арифметические выражения с переменными и константами, индексные выражения, операторы присваивания, составные операторы, условные операторы, циклические операторы, операторы ввода-вывода, а также описание переменных и массивов.

Методика выполнения:

Теория описана в [1], разд. 4.1, 4.3 (пример 5), 5.4 (пример 14), 5.5 (примеры 15, 16,17), 5.6 (пример 18), 5.7 (пример 19).

Практическое занятие № 5

Преобразование контекстно-свободной грамматики к обобщенной нормальной форме Грейбах.

Цель занятия:

Научиться анализировать порождающие правила контекстно-свободной грамматики и преобразовывать их к обобщенной нормальной форме Грейбах.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов преобразования порождающих правил контекстно-свободной грамматики к обобщенной нормальной форме Грейбах.

1.2. Освоение навыков анализа процесса порождения цепочек контекстно-свободной грамматикой, заданной в виде обобщенной нормальной форме Грейбах.

Задание:

Преобразовать контекстно-свободную грамматику, разработанную на 4-м занятии к обобщенной нормальной форме Грейбах и проанализировать множество порождаемых ею цепочек.

Методика выполнения:

Теория описана в [1], разд. 4.4 (пример 6), 4.5 (пример 7), 5.4 (пример 14), 5.5 (примеры 15, 16,17), 5.6 (пример 18), 5.7 (пример 19).

Практическое занятие № 6

Построение и реализация LL(1)-анализатора.

Цель занятия:

Научиться разрабатывать и реализовывать в виде программы LL(1)-анализатор, построенный по порождающим правилам контекстно-свободной грамматики, заданной в обобщенной нормальной форме Грейбах.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение метода построения LL(1)-анализатора по порождающим правилам контекстно-свободной грамматики, заданной в обобщенной нормальной форме Грейбах.

1.2. Получение навыков реализации LL(1)-анализатора в виде программы на языке программирования, вызывающей лексический анализатор для выделения очередной лексемы.

1.3. Получение навыков тестирования LL(1)-анализатора.

Задание:

Разработать, реализовать в виде программы и протестировать LL(1)-анализатор, построенный по контекстно-свободной грамматике, преобразованной к обобщенной нормальной форме Грейбах на 5-м занятии, и вызывающий лексический анализатор для выделения очередной лексемы, реализованный на 3-м занятии.

Методика выполнения:

Теория описана в [1], разд. 4.6 (пример 8), 5.2 (пример 11), 5.4 (пример 14), 5.5 (примеры 15, 16, 17), 5.6 (пример 18), 5.7 (пример 19).

Практическое занятие № 7

Построение семантической таблицы в LL(1)-анализаторе для генерации обратной польской строки.

Цель занятия:

Научиться разрабатывать семантическую таблицу для генерации обратной польской строки в LL(1)-анализаторе, построенном по порождающим правилам контекстно-свободной грамматики, заданной в обобщенной нормальной форме Грейбах.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение метода построения семантической таблицы для генерации обратной польской строки в LL(1)-анализаторе.

1.2. Получение навыков анализа генерируемых последовательностей обратной польской строки при работе LL(1)-анализатора.

Задание:

Разработать и проанализировать семантическую таблицу для генерации обратной польской строки в LL(1)-анализаторе, построенном на 6-м занятии.

Методика выполнения:

Теория описана в [1], разд. 5.2 (пример 11), 5.4 (пример 14), 5.5 (примеры 15, 16, 17), 5.6 (пример 18), 5.7 (пример 19).

Практическое занятие № 8

Построение и реализация генератора обратной польской строки в виде LL(1)-анализатора.

Цель занятия:

Научиться разрабатывать и реализовывать в виде программы генератор обратной польской строки в виде LL(1)-анализатора.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение метода построения генератора обратной польской строки в виде LL(1)-анализатора.

1.2. Получение навыков реализации генератора обратной польской строки, как LL(1)-анализатора в виде программы на языке программирования.

1.3. Получение навыков реализации семантических программ генерации обратной польской строки.

1.4. Получение навыков тестирования генератора обратной польской строки, реализованного как LL(1)-анализатора.

Задание:

Разработать, реализовать в виде программы и протестировать генератор обратной польской строки на основе LL(1)-анализатора, построенного на 6-м и 7-м занятиях.

Методика выполнения:

Теория описана в [1], разд. 5.2 (пример 11), 5.4 (пример 14), 5.5 (примеры 15, 16,17), 5.6 (пример 18), 5.7 (пример 19).

Практическое занятие № 9

Построение и реализация интерпретатора обратной польской строки.

Цель занятия:

Научиться разрабатывать и реализовывать в виде программы интерпретатор обратной польской строки.

Составляющие практических навыков и приемов, изучаемых на занятии:

1.1. Изучение методов построения интерпретатора обратной польской строки.

1.2. Получение навыков реализации интерпретатора обратной польской строки в виде программы на языке программирования.

1.3. Получение навыков реализации операций в интерпретаторе обратной польской строки.

1.4. Получение навыков тестирования интерпретатора обратной польской строки.

Задание:

Разработать, реализовать в виде программы и протестировать интерпретатор обратной польской строки, генерируемой LL(1)-анализатором, построенном на 8-м занятии.

Методика выполнения:

Теория описана в [1], разд. 5.1 (примеры 9, 10), 5.2 (пример 11), 5.3 (пример 12, 13), 5.4 (пример 14), 5.5 (примеры 15, 16), 5.6 (пример 18), 5.7 (пример 19).

Библиографический список

1. Костюк Ю.Л. Конспект лекций по дисциплине «Методы трансляции».
2. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. В двух томах / Пер. с англ. М: Мир, 1978.
3. Ахо А., Сети В., Лам М., Ульман Дж. Компиляторы: принципы, технологии и инструментарий / Пер. с англ. М: Вильямс, 2008.
4. Вирт Н. Построение компиляторов / Пер. с англ. М: ДМК Пресс, 2010.
5. Мартыненко Б.К. Языки и трансляции. СПб: Изд-во С-Петербург. ун-та, 2004.