

К экзамену

Вопросы к экзамену

1. Определение оконченного автомата. Синтез автомата.
2. Из чего состоит конечный автомат? Способы представления
Диаграмма состояний конечного автомата. Матрицы переходов конечного автомата.
3. Чем отличается автомат Мили от автомата Мура?
4. Сформулируйте основные действия по преобразованию автомата Мили в автомат Мура. Примеры.
5. Сформулируйте основные действия по преобразованию автомата Мура в автомат Мили. В чем заключается целесообразность использования двух различных типов конечных автоматов?
6. Какие автоматы называют эквивалентными?
7. алгоритм минимизации конечного автомата.
8. Как определить, являются ли конечные автоматы эквивалентными?
9. Недетерминированные конечные автоматы как распознаватели.
10. Регулярные выражения. Правила преобразования.

1. Детерминированный конечный автомат
2. Эквивалентность недетерминированных и детерминированных конечных автоматов
3. Регулярные языки
4. Лемма накачки регулярных языков
5. Свойства замкнутости регулярных языков
6. Грамматики. Классификация Хомского.
7. Регулярные грамматики.
8. Регулярные грамматики и недетерминированные конечные автоматы восходящий и нисходящий разбор.
9. МП- автомат. МП- распознаватели КС- языков.
10. Однозначные и неоднозначные МП- автоматы.
11. КС- грамматики. Приведение грамматик.
12. Нормальная форма Хомского.
13. Алгоритм Кока-Янгеля-Косами.

Литература

1.	А. Ахо. Д. Ульман.	Теория синтаксического анализа, перевода и компиляции. М.:Мир,	1978.	140	4
1.	. Хопкрофт Д., Мотвани Р., Ульман Д.	Введение в теорию автоматов, языков и вычислений. Изд. дом "Вильямс", 2011, 527 с. проектированию	2011	60	4
1.	Хантер	Проектированиеи конструирование компиляторов			
1.	Ахо А., Сети Р., Ульман Дж. Д.	Компиляторы: принципы, технологии и инструменты. М.: Вильямс, 2001.	2001		
ДОПОЛНИТЕЛЬНАЯ					
1.	Саломая А.	Жемчужины теории формальных языков. М.: Мир, 1986	1986	10	2

Лабораторные работы

- 1 Необходимо перевести автомат из/в Милли/Мура в зависимости от того что подается как входной файл.
- 2 Необходимо минимизировать входной автомат.
- 3 Преобразовать НКА в ДКА. Только для автоматов Милли.
- 4 Исключить не продуктивные и не достижимые правила входной КС грамматики. Подробное описание алгоритма находится в Л-7 - Приведение грамматик.
- 5. Реализовать алгоритм Кока - Янгера - Касами для КС грамматики с демонстрацией его работы через построение таблицы разбора. Подробное описание алгоритма находится в Л-8 - Алгоритм Кока

РГР

- РГР в 1-м семестре это часть разработки комплексного проекта «Мой первый компилятор» в рамках курса ТАиФЯ.
- Цель в этом семестре разработать основы языка программирования и часть компилятора – лексический анализатор.

ОСНОВНЫЕ ЗАДАЧИ РГР

1. Разработать требования к создаваемому языку
2. Определить алфавит языка, основные лексические единицы
3. Определить принципы работы лексического анализатора- сканера
4. Привести грамматику языка с учетом лексических элементов языка
5. Используя генератор синтаксических распознавателей проверить грамматику языка на приведенность и однозначность
6. Полученный модуль распознавателя проверить на тестах и описанных в п.1 примерах решаемых задач.
7. Оформить проделанную работу в соответствии с требованиями к РГР

Разработать требования к создаваемому языку

- 1. Определить назначение языка
- 2. Привести примеры решаемых задач
- 3. Определить приоритетные свойства языка

Определить алфавит языка, основные лексические единицы

- Основные лексические единицы (токены)- идентификаторы, числа, знаки операций, разделители, статус пробела и т.д.
- Привести примеры лексем.
- Привести грамматику и регулярное выражение для токенов

Определить принципы работы лексического анализатора

- Сканер должен работать в составе генератора, следовательно, она по интерфейсам согласовываться с модулем разбирающим грамматику.
- Изначально определим:
- Вход:
- Анализируемая программа считывается из файл с расширением txt, содержит текст программы
- Выход:
- Выходом сканера является необходимая для генератора информация.
- дополнительный выход , содержит список лексем и соответствующие токены(класс, тип).

Описать структуру программы, типы используемых данных.
Описать достоинства и недостатки вашего подхода.

Используя генератор синтаксических распознавателей проверить грамматику языка на приведенность и однозначность

- Разобраться в выходной информации генератора.
- Определить и привести в РГР выходные данные, доказывающие правильность грамматики языка и, соответственно, сканера.

Полученный модуль распознавателя проверить на описанных выше примерах решаемых задач.

- Написать «чистые» и «грязные» тесты
- Проверить работу распознавателя
- Результаты представить в РГР

Оформить проделанную работу в соответствии
с требованиями к РГР.
Публичная защита работ 24.04

