

Содержание

Занятие 1.	3
Теория.....	3
Проект.....	3
Практика	3
Задание	3
Занятие 2.	3
Теория.....	3
Практика	4
Задание на занятие	4
Задание на семестр	4
Задание на дом.....	4
Занятие 3.	5
Теория.....	5
Практика	5
Задание	5
Занятие 4.	5
Проект.....	5
Теория.....	5
Практика	5
Задание	5
Занятие 5.	6
Проект.....	6
Теория.....	6
Практика	6
Задание	6
Занятие 6.	6
Проект.....	6
Теория.....	6
Практика	6
Задание	6
Занятие 7.	6
Проект.....	6
Теория.....	6

Практика	7
Задание	7
Занятие 8.	7
Проект.....	7
Теория.....	7
Практика	7
Задание	7
Занятие 9.	7
Проект.....	7
Теория.....	7
Практика	7
Задание	7
Занятие 10.	7
Проект.....	7
Теория.....	7
Практика	7
Задание	8
Занятие 11.	8
Проект.....	8
Теория.....	8
Практика	8
Задание	8
Занятие 12.	8
Проект.....	8
Теория.....	8
Практика	8
Задание	8
Занятие 13.	8
Проект.....	8
Теория.....	8
Практика	8
Задание	9
Тесты	9

Занятие 1.

Теория

Цель курса.

Введение в теорию языков, работа с проектом, разбор проекта. Дополнение проекта.

История развития трансляторов.

Ассемблер. Мнемокод. Интерпретатор. Компилятор. Объектный язык. Статическая типизация, динамическая типизация. Системы построения компиляторов. Среда разработки. GCC, VC, NetBeans, Eclipse, LLVM.

Архитектура трансляторов.

Сканер, Синтаксис, Семантика, Оптимизация программы, Распределение памяти, Генерация кода, Оптимизация кода, Компоновщик, Загрузчик, Стековая машина.

Проект

Структура проекта

TranParse, TranSyntax, TranRules, TranGrammar, TableBuilder, TranBuilder, TranYACC, Translator, TreeNodeCode.

Задания. Лексика, Проект, Стековая машина, Грамматика.

Практика

Знакомство с проектом. Поиск ошибок. Поиск мест выполнения задач. Разбор задач.

Задание

Вторая глава книги дракона.

Сканер. TranParse.

Занятие 2.

Теория

Scan и его окружение. Сам простой, сложности в использовании. Откаты. Пропуски незначимых.

На входе. Source (File, String). Стек(Source), Array(File)

На выходе. Parse (Array, Syntax). Группировка по приоритетам, типам. Управляемый и неуправляемый сканеры. Контекст.

```
enum T_Group { grNone, grOper, grSeparator, grInteger, grNumber, grIdent, grReserv, grSpace, grLine, grError, grPriority, grSyntax, grEof };
```

```
enum T_Syntax { sxNone, sxIdentFunc, sxIdentArr, sxDefFunc, sxDefArr, sxDefVar, sxDefType, sxLabel, sxCast, sxCastUnary, sxEmptyCramp, sxEmptyBracket, sxEmptyScrape };
```

```
enum W_Type { rwNone, rwFor, rwWhile, rwIf, rwElse, rwDo, rwGoTo, rwContinue, rwBreak, rwReturn,
rwSwitch, rwCase, rwTry, rwThrow, rwCatch, rwEnum, rwDefault, rwVoid, rwShort, rwInt, rwLong, rwFloat,
rwDouble, rwDefine, rwConst, rwStatic, rwAuto, rwChar, rwExtern, rwRegister, rwSigned, rwSizeof,
rwStruct, rwTypedef, rwUnion, rwUnsigned, rwVolatile };
```

```
enum T_Priority { prNone, prAssign, prQuestion, prOrOr, prAndAnd, prOr, prXOr, prAnd, prEqu, prCompare,
prShift, prTerm, prFactor, prUnary, prIncDec };
```

Внутри. Таблица идентификаторов. Таблица строк. Таблица констант.

Отдельная фаза. Микро и макро мир. Регулярные выражения и конечные автоматы. Lex – сканер.

Практика

TranParse, TranSyntax, TranRules, TranGrammar, TableBuilder, TranBuilder, TranYACC, Translator, TreeNodeCode.

Задание на занятие

- HEX – числа. Сканер разбирает только целые десятичные и восьмеричные числа и десятичные с фиксированной точкой. Нет шестнадцатеричных, двоичных и с плавающей точкой.
- Комментарий. /* */ Много строчный комментарий.

Задание на семестр

- #include – Сканер работает только со строкой в памяти и с одним файлом. Желательно реализовать аппарат подключаемых файлов. Макросы. Аппарат макрокоманд без параметров очень просто реализуется с использованием аппарата #include.
- Макросы с параметрами
- #ifdef – условная компиляция без логических условий. #if defined() - условная компиляция с логическими условиями && и ||.
- Использование FLEX (генерация шаблона, Priority, Syntax)
- SQL. Построение дерева.
- Флора OTL, F++.
- Грамматика NL.
- Реализация оператора switch
- Левосторонняя рекурсия. Empty – правила. Левая факторизация
- Форма Хомского
- Алгоритм Кока-Янгера-Касами
- Другой язык.

Задание на дом

Третья глава книги дракона.

Занятие 3.

Теория

Подключение грамматики C. YACC, Lex. Строковые константы, type_name, ключевые слова, “->”, “...”, константы. Контекст сканера (type_name, “=”, “&”). Не работает без синтаксиса.

Прямой нисходящий рекурсивный спуск. Z(), T(), F().

Грамматика. Формы записи. Левосторонняя и правосторонняя рекурсии. Построение программ и распознавателей по грамматике.

Практика

Сделать. HEX – числа. Строковые данные, преобразование тела и его хранение. Много строчный комментарий.

Оценки.

Задание

Классы T_Syntax и TSyntax.

Четвертая глава книги дракона.

Занятие 4.

Проект

Разбор Грамматики C. Проблема с TYPE_NAME. Таблица блоков, таблица объявлений. Опережающее чтение лексем.

Теория

Стековые распознаватели для нисходящего и восходящего разборов. Начальное состояние стеков. Стек уровней и разбор правил. И то и то в тексте программ. Первое в стеке вызовов второе интерпретация в программе. Прямая рекурсия Formula, Produce. Первое в стеке второе в интерпретация в теле программы. И то и то в стеке. Распознаватель ParseStack . Рекурсия с раскрытием продукций в стеке. Рекурсия в стеке символов.

Нисходящий - знание(предположение) правила и его раскрытие. Восходящий - свертка правила в стеке. Незнание приводит к откатам.

Откаты сканера и синтаксиса. Порядок правил. Распределение отката между сканером и синтаксисом (Priority, Syntax). Откат входа и откат выхода. Условие безоткатности.

Динамическое программирование.

Практика

Задание

Четвертая глава книги дракона.

Занятие 5.

Проект

Разбор Грамматики С. Проблема с TYPE_NAME. Таблица блоков, таблица объявлений. Опережающее чтение лексем.

Действия. Представление программы. Виды синтаксиса. Преобразование формы записи грамматик.

Теория

Функции FIRST, FOLLOW, GOTO, CLOSURE.

Распознаватели на основе таблиц автомата со стековой памятью.

Практика

Задание

Четвертая глава книги дракона.

Занятие 6.

Проект

Разбор Грамматики С. Проблема с TYPE_NAME. Таблица блоков, таблица объявлений. Отслеживание состояний разборщика. VerifyAction (IDENTIFIER , TYPE_NAME, DECLARATOR , ENUMERATOR)

Действия. Представление программы. Дерево программы. Построение дерева снизу-вверх, даже для рекурсивного спуска.

Теория

Функции FIRST, FOLLOW, GOTO, CLOSURE.

Практика

Example_00 – Ассоциативность левая и правая. Дерево. Коды. Операторы, функции. Примеры.

Example_01 – Рекурсия в стеке и итерация.

Example_02 – FIRST, FOLLOW.

Example_03 – C99_Grammar. Примеры. Создание примера.

Задание

Четвертая глава книги дракона.

Занятие 7.

Проект

Действия. Действия в стеке при нисходящем разборе. По таблице.

Теория

Утрированные высказывания. Атрибуты узлов дерева. Синтетические и наследуемые атрибуты.

Практика

Example_00 – Ассоциативность левая и правая. Дерево. Коды. Операторы, функции. Примеры.

Example_01 – Рекурсия в стеке и итерация.

Example_02 – FIRST, FOLLOW.

Example_03 – C99_Grammar. Примеры. Создание примера.

Задание

Пятая глава книги дракона.

Занятие 8.

Проект

Действия в стеке при нисходящем разборе. По правилам.

Теория

Работа с типами. Обход дерева и таблица блоков.

Практика

Задание

Пятая глава книги дракона.

Занятие 9.

Проект

Передача таблиц и грамматики как исходный текст.

Теория

Генерация кодов.

Практика

Задание

Занятие 10.

Проект

Отключение оптимизации при трансляции определений.

Теория

Генерация кодов. Логические выражения. Переход из одного вида в другой.

Команды перехода. Вперед - назад. Метка переход.

Практика

Example_05 – Создание переменных и массивов.

Example_06 – Логические выражения.

Example_07 – Распределение памяти. Виды памяти

Example_08 – Реализация операторов.

Example_09 – Break, continue. Ошибки.

Example_10 – Goto, label.

Задание

Занятие 11.

Проект

ETF.bat

Теория

FIRST, FOLLOW. Таблицы нисходящего разбора.

Практика

ETF.bat, LL1 – таблицы.

Задание

Занятие 12.

Проект

Преобразование грамматик. Переход к лексемам. Левая факторизация.

Теория

GOTO, CLOSURE. Таблицы восходящего разбора. Пункты.

Нисходящий разбор. История разбора. Состояние автомата.

Практика

ETF.bat, LR – таблицы.

Задание

Занятие 13.

Проект

Попытка расширения Core(A -> empty). Проверка Core (FIRST == terminals).

Проверка LRTable == GOTO.

Теория

Практика

Тесты.

Грамматики.

Задание

Занятие 14.

Проект

Теория

Практика

Тесты.

Грамматики.

Задание

Занятие 15.

Проект

LRTable сжатие.

Теория

Практика

Задание

Тесты

Проверка на правильность и на ошибки. Операторы в теле функций.

Переменные

Массивы

Блоки и область видимости

Операторы управления

Операторы перехода и метки пользователя.

Заменители goto

Функции

Инициализация переменных и массивов.

Арифметические выражения

Логические выражения

Условия