

## Обзор

### История

- Личная
  - ТА-2 (БЭСМ-4, 20 КГц)
  - Две области точек
  - Конвертор МИР-ЕС(Fortran)
  - Флора (F+, стековая машина, объектная машина)
- Преподавание
  - МГМИ (Алгол-60, зеркальное отражение)
  - Лицей (Информатика)
  - МГМИ (Транслятор - 70 Кб, 1-4 группа)
  - Школа №5 (Калькулятор)
  - ИТФИ

### Проект

- Цель
- Структура

### Лего

#### Структуры

- Лексема
- Массив лексем
- TScan, TParse
- Таблица грамматики
  - Формула
  - Продукция
  - Элемент (терминал-лексема, формула, продукция, ...)
- Абстрактное синтаксическое дерево AST
- Вектор
- Таблица векторов
- Таблица LL(k) разбора
  - Cell
- Таблица LR(k) разбора
  - Action
  - Situation
  - Para
  - ParaSet

### Модули

- Формирование текстов структур данных
  - Заголовок
  - Перечисление
  - Таблица символов
  - Таблица ключевых слов

- Получение файла лексем простого текста простым сканером
  - Простые лексемы
  - Составные лексемы
  - Лексемы приоритетов
- Сканер YACC строковые константы, шестнадцатеричные константы и многострочные комментарии. Выход файл лексем
- Сканирование текста C и C99 с лексемами TYPE\_NAME и IDENTIFIER. Выход файл лексем
- Чтение файла лексем с формированием массива лексем с набором текстовых значений
- Чтение таблицы грамматики
- Печать массива лексем
- Вывод массива лексем
- Печать таблицы грамматики
  - В виде YACC
  - В нормальной форме Бэкуса
  - В итерационной форме Бэкуса
- Вывод таблицы грамматики
- Синтаксис LL1\_ETF
- Синтаксис BIF\_ETF
- LL1 разбор рекурсивными функциями *Формула* и *Продукция* при обходе *Таблицы грамматики*.
  - Формирование AST
- LL1 разбор итерационной функцией со стеком при обходе *Таблицы грамматики*.
  - Формирование AST
- Таблицы разбора
  - Вывод
    - Таблицы LL(k) разбора
    - Таблицы LR(k) разбора
  - Чтение
    - Таблицы LL(k) разбора
    - Таблицы LR(k) разбора
  - Печать
    - Таблицы LL(k) разбора
    - Таблицы LR(k) разбора

Сканер

Parse.h, Parse.cpp

Есть много вариантов организации работы сканера, как по входному и выходному потокам, так и по структуре программы.

Обрабатывает весь текст за раз

Обрабатывает текст лексема за лексемой

Читает текст из файла \*.cpp

Читает текст из строки

Реализует откат позиции входного текста, либо с установкой произвольной позиции, либо откат на один символ

Формирует массив лексем

Пишет лексемы в файл \*.lex

Передает лексемы синтаксису одна за одной

Передает текущую лексему, какого типа она бы не была, со сдвигом позиции в исходном тексте

Передает текущую лексему, если ее тип совпадает с запрашиваемым типом, и сдвигает позицию в исходном тексте

Структура

Читает файл в строку, затем эту строку разбирает.

Использует мини-сканер, который читает символ входного потока и выдает его тип: буква, цифра, символ, конец строки и конец файла. Это облегчает сбор чисел, идентификаторов и ключевых слов.

Есть таблица символов (T\_Symbol) (scanner.sym) и таблица ключевых слов (T\_Word) (scanner.key).

Есть функции сбора числа, сбора идентификатора, проверки ключевого слова, поиска операции и разделителя.

После сборки идентификатора поиск в таблице ключевых слов

Сбор комментария однострочного и многострочного

Пропуск пробелов и концов строк

Задачи

Чтение файла \*.lex с печатью

Чтение файла \*.lex с записью в массив лексем

Формирование псевдосканера Scan

- массив лексем внутри
- курсор на текущую лексему, при старте на первую
- методы match(группа лексемы, тип лексемы). Если текущая совпадает - смещение курсора и возврат истина, иначе лож.

Создание мини-синтаксиса ZETF. TranParse.cpp

- Подключение псевдосканера Scan
- Чтение файла test.lex с записью в массив лексем
- Проверка синтаксиса

Создание полноценного сканера

