

НИУ ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1
по дисциплине
«Системное программное обеспечение»
Вариант 1

Выполнил: студент группы Р4116

Крюков Андрей

Преподаватель: Кореньков Юрий Дмитриевич

Санкт-Петербург
2023 г.

Задание:	3
Описание структур данных	4
Примеры входных данных и результаты их обработки	4
Вывод	5

Задание:

Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора текста в соответствии с языком по варианту. Реализовать построение по исходному файлу с текстом синтаксического дерева с узлами, соответствующими элементам синтаксической модели языка. Вывести полученное дерево в файл в формате, поддерживающем просмотр графического представления.

Порядок выполнения:

1. Изучить выбранное средство синтаксического анализа
 - a. Средство должно поддерживать программный интерфейс, совместимый с языком Си
 - b. Средство должно параметризоваться спецификацией, описывающей синтаксическую структуру разбираемого языка
 - c. Средство может функционировать посредством кодогенерации и/или подключения необходимых для его работы дополнительных библиотек
 - d. Средство может быть реализовано с нуля, в этом случае оно должно использовать обобщенный алгоритм, управляемый спецификацией
2. Изучить синтаксис разбираемого по варианту языка и записать спецификацию для средства синтаксического анализа, включающую следующие конструкции:
 - a. Подпрограммы со списком аргументов и возвращаемым значением
 - b. Операции контроля потока управления – простые ветвления if-else и циклы или аналоги
 - c. В зависимости от варианта – определения переменных
 - d. Целочисленные, строковые и односимвольные литералы
 - e. Выражения численной, битовой и логической арифметики
 - f. Выражения над одномерными массивами
 - g. Выражения вызова функции
3. Реализовать модуль, использующий средство синтаксического анализа для разбора языка по варианту
 - a. Программный интерфейс модуля должен принимать строку с текстом и возвращать структуру, описывающую соответствующее дерево разбора и коллекцию сообщений об ошибке
 - b. Результат работы модуля – дерево разбора – должно содержать иерархическое представление для всех синтаксических конструкций, включая выражения, логически представляющие собой иерархически организованные данные, даже если на уровне средства синтаксического анализа для их разбора было использовано линейное представление
4. Реализовать тестовую программу для демонстрации работоспособности созданного модуля
 - a. Через аргументы командной строки программа должна принимать имя входного файла для чтения и анализа, имя выходного файла записи для дерева, описывающего синтаксическую структуру разобранного текста
 - b. Сообщения об ошибке должны выводиться тестовой программой (не модулем, отвечающим за анализ!) в стандартный поток вывода ошибок
5. Результаты тестирования представить в виде отчета, в который включить:
 - a. В части 3 привести описание структур данных, представляющих результат разбора текста (3а)
 - b. В части 4 описать, какая дополнительная обработка потребовалась для результата разбора, предоставляемого средством синтаксического анализа, чтобы сформировать результат работы созданного модуля
 - c. В части 5 привести примеры исходных анализируемых текстов для всех синтаксических конструкций разбираемого языка и соответствующие результаты разбора

Описание структур данных

В ходе парсинга формируется дерево, представленное следующей структурой.

```
typedef struct ASTNode ASTNode;  
  
struct ASTNode {  
    char *type;  
    ASTNode *left;  
    ASTNode *right;  
    char *value;  
    int id;  
};
```

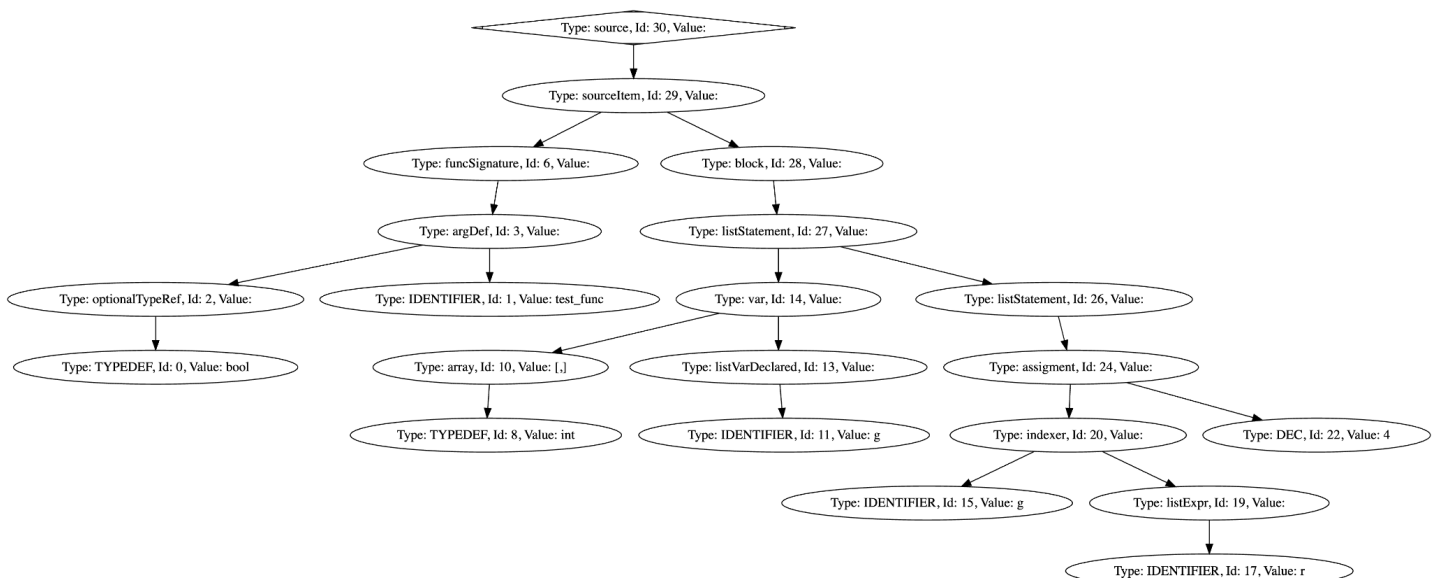
При исполнении правила bison, выполняется код, создающий очередную ноду дерева

```
funcSignature: argDef LPAREN listArgDef RPAREN {{ $$ = createNode("funcSignature", $1,  
$3, ""); }}
```

После завершения парсинга, дерево при помощи рекурсивного алгоритма представляется в виде графа в формате dot, после чего его можно визуализировать при помощи, к примеру, graphviz.

Примеры входных данных и результаты их обработки

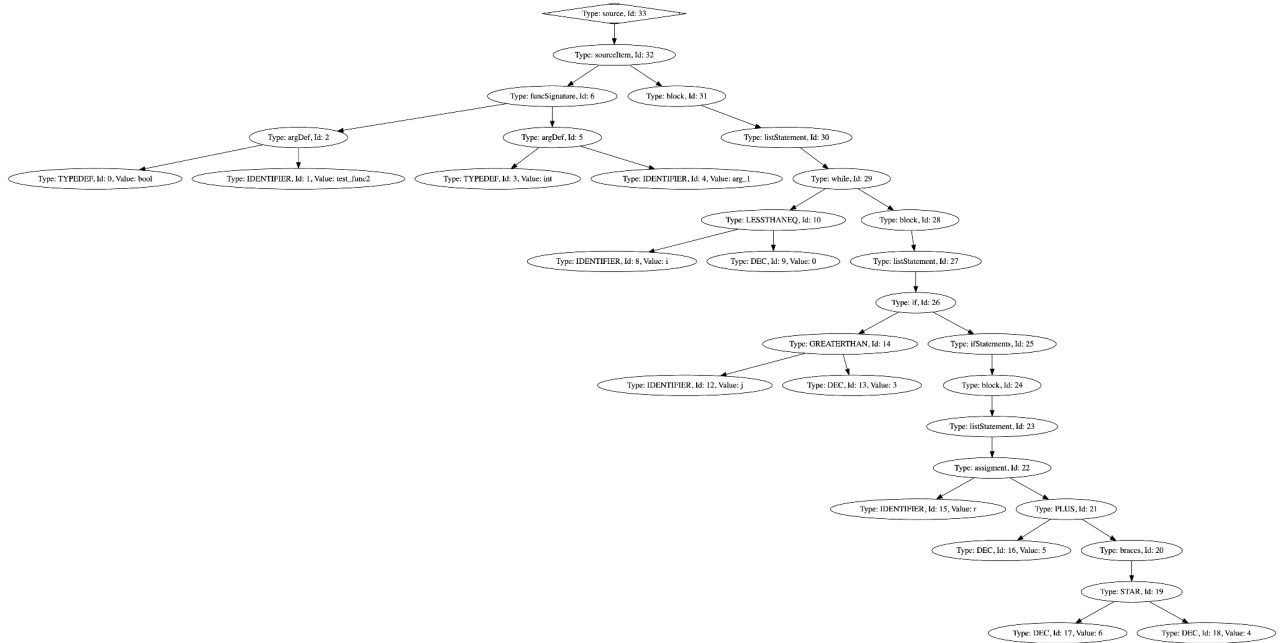
```
bool test_func() {  
    int[,] g;  
    g[r] = 4;  
}
```



```

bool test_func2 (int arg_1 ) {
    while (i <= 0) {
        if (j > 3) {
            r = 5;
        }
    }
}

```



Вывод

В ходе данной лабораторной работы был имплементирован лексер, парсер и алгоритмы построения и вывода в формате .dot абстрактного синтаксического дерева для текста на языке, соответствующем заданной по варианту формальной грамматике.