

## Задание практикума «Анализатор грамматик в формате GNU bison»

Нужно написать программу, которая по текстовому файлу с грамматикой описанной в формате GNU bison (<http://www.gnu.org/software/bison/>) выдаст ответ: применим ли к данной грамматике метод рекурсивного спуска или нет. Необходимо так же написать программу, которая должна приводить грамматику к каноническому виду удаляя недостижимые и бесплодные символы. Результатом работы должен быть текстовый файл с описанием грамматики в формате GNU bison, а так же программу, которая удаляет из грамматики epsilon альтернативы, строя при этом эквивалентную исходной грамматику.

Должны быть реализованы как библиотечные функции обязательные компоненты:

1. Лексический анализатор – должен разбирать файл в формате GNU bison на лексемы.
2. Синтаксический анализатор – должен определять корректность синтаксиса файла в формате GNU bison. Требуется реализация методом рекурсивного спуска.
3. Семантический анализатор – должен позволять записывать грамматику во внутренний формат в оперативную память для дальнейшей её обработки. Некоторые функции семантического анализа можно совместить с синтаксическим.

Требования к ходу выполнения задания:

1. Необходимо описать контекстно свободную грамматику, которая описывает текстовый формат файлов GNU bison, и при этом допускает разбор методом рекурсивного спуска. В качестве терминального символа используется лексема. Лексемы получаются с этапа лексического анализа. Для этапа лексического анализа также необходимо описать регулярную грамматику.
2. Пишется программа, которая умеет отвечать на файл с расширением .u или .urr (**Да** – файл синтаксически верен и **Нет** – была синтаксическая ошибка в такой-то строке и в таком-то столбце).
3. В грамматику вставляются действия, которые при разборе формируют в памяти процесса хранилище правил грамматики. (В этом месте можно использовать STL.)
4. К данным, полученным в результате разбора и хранящимся в оперативной памяти применяете алгоритмы анализа грамматики, реализуя их в двух программах.

Далее приведёно некоторое упрощённое описание синтаксиса bison файла.

Файл начинается со служебной информации. Служебная информация начинается с декларации конструкций языка C. Конструкции описываются внутри скобок организованных следующим образом: `%{ /* Здесь код */ }%`. Далее в качестве служебной информации выступает описание лексем в bison грамматике. **«%token Имя\_лексемы;»** (в тексте могут встречаться так же одно символьные не именованные лексемы, например '\r' или 'f'). Далее после %% идёт собственно описание грамматики.

Правила выглядят следующим образом: **«левая\_часть: первая\_альтернатива | вторая\_альтернатива | третья\_альтернатива ;»**. Здесь левая часть является не терминальным символом контекстно-свободной грамматики, а альтернативы – это один из вариантов порождаемой цепочки. Возможно указание пустой цепочки, тогда в альтернативе не будет присутствовать ни одной лексемы и ни одного не терминального символа.

В файле могут встречаться комментарии в стиле языка программирования C. Текст указанный в фигурных скобках используется для описания действий в правой части правил контекстно-свободной грамматики. Описания правил заканчиваются двумя %%. Далее идёт код на языке C/C++, который с точки зрения данного задания должен восприниматься просто как комментарий.

Ниже приведён пример файла с описанием КС грамматики:

```
%{
#include <stdio.h>
%}

%token NUM;

%%
input:      /* empty */
           | input line
           ;

line:       '\n'
           | exp '\n'      { printf ("\t%.10g\n", $1); }
           ;

exp:        NUM            { $$ = $1;                }
           | exp exp '+'   { $$ = $1 + $2;          }
           | exp exp '-'   { $$ = $1 - $2;          }
           | exp exp '*'   { $$ = $1 * $2;          }
           | exp exp '/'   { $$ = $1 / $2;          }
           /* Exponentiation */
           | exp exp '^'   { $$ = pow ($1, $2); }
           /* Unary minus */
           | exp 'n'       { $$ = -$1;              }
           ;
%%
```

Более подробно о том как и что может содержаться в файле с грамматиками можно прочитать вот здесь:

[http://www.gnu.org/software/bison/manual/html\\_node/index.html](http://www.gnu.org/software/bison/manual/html_node/index.html)

[http://trpl.narod.ru/CC\\_Bison.htm](http://trpl.narod.ru/CC_Bison.htm)