

Лабораторная работа №3. Использование автоматических генераторов анализаторов Bison и ANTLR

10 апреля 2013 г.

Введение

Цель данной лабораторной работы — научиться пользоваться автоматическими генераторами анализаторов Bison и ANTLR.

Форма отчетности: программа и текстовый отчет. Средство автоматической генерации вы можете выбрать самостоятельно.

Рекомендуемые источники:

<http://www.gnu.org/software/bison> — Bison

<http://www.antlr.org> — ANTLR

Задания

Вариант 1. Выражения в префиксной записи

Трансляция выражения из префиксной записи в язык высокого уровня.

На входе — выражение в префиксной записи, должны поддерживаться арифметические и логические операции, операции сравнения чисел, ветвление, присваивание, вывод.

На выходе программа на императивном языке высокого уровня.

Пример:

```
if > 2 3 print 3 if > 4 - 7 2 print + 3 4
```

Вывод:

```

begin
  if (2 > 3) then
    writeln(3)
  else if (4 > (7 - 2)) then
    writeln(3 + 4);
end.

```

Вариант 2. Арифметические выражения

Вычисление арифметических выражений с заведением переменных. В результате трансляции должно вычисляться значение выражений, в выражении допускается присваивание значений переменных. Используйте целочисленные переменные.

Пример:

```

a = 2;
b = a + 2;
c = a + b * (b - 3);
a = 3;
c = a + b * (b - 3);

```

Вывод:

```

a = 2
b = 4
c = 6
a = 3
c = 7

```

Вариант 3. Перевод с Паскаля на Си

Выберите подмножество языка Паскаль и напишите транслятор, который переводит программы на заданном подмножестве на язык Си.

Вы можете выбрать небольшое подмножество языка, но на входе и на выходе вашего транслятора должны быть компилирующиеся программы.

Пример:

```

var
  a, b : integer;
begin
  read(a, b);

```

```
writeln(a + b);  
end.
```

Вывод:

```
int a, b;  
  
int main() {  
    scanf("%d%d", &a, &b);  
    printf("%d\n", a + b);  
    return 0;  
}
```

Вариант 4. Функциональный язык

Придумайте примитивный функциональный язык программирования и реализуйте его трансляцию в императивный язык.

Пример:

```
fac :: Integer -> Integer  
fac 0 = 1  
fac n | n > 0 = n * fac(n - 1)
```

Вывод:

```
function fac(n: integer): integer;  
begin  
    if n = 0 then  
        fac := 1  
    else if n > 0 then  
        fac := n * fac(n - 1);  
    end;  
end;
```

Вариант 5. Идеальное форматирование

Выберите подмножества языка C++, Java или Pascal и напишите преобразование программы на заданном подмножестве этого языка в идеально отформатированную программу. Для Java следует использовать Java Coding Conventions. Для C++ вы можете выбрать способ форматирования на свое усмотрение, например Google C++ Style Guide. Для Паскаля можете выбрать разумный аналог.

Пример:

```

int
    main( )
{
    printf( "Hello  world"
    )
; return  0
;}

```

Вывод:

```

int main() {
    printf("Hello  world");
    return 0;
}

```

Вариант 6. Обфускация

Выберите подмножества языка C++, Java или Pascal и напишите обфускатор для программ данного подмножества. Обфускатор должен заменять имена переменных на случайные строки из символов I, 1, O и 0, которые являются корректными идентификаторами и в случае одинаково выглядящих символов I – 1 и O – 0, соответственно, выглядеть одинаково. Также обфускатор должен вставлять в различные места программы незначащие действия с переменными, которые затрудняют понимание программы, в том числе добавлять новые переменные.

Пример:

```

var
    a, b : integer;
begin
    read(a, b);
    writeln(a + b);
end.

```

Вывод:

```

var
    I010, IOIO : integer;
    IO10 : integer;
begin
    read(IO10, IOIO);
    IO10 := IOIO;

```

```
writeln(I0I0 + IOIO);  
end.
```

Вариант 7. Хороший язык

Придумайте хороший императивный язык программирования, на котором приятно писать программы. Транслируйте с него в Си.

Пример:

```
a, b = readint(), readint()  
a, b = b, a  
print(a + b)
```

Вывод:

```
int main() {  
    int a, b;  
    scanf("%d%d", a, b);  
    int ta = a;  
    int tb = b;  
    a = tb;  
    b = ta;  
    printf("%d", a + b);  
}
```

Вариант 8. TEX2HTML

Выберите подмножество тека и напишите его конвертор в HTML. При необходимости используйте MathML.

Пример:

```
$a_i = b_i + x^2$
```

Вывод:

```
<i>a</i><sub><i>i</i></sub> = <i>b</i><sub><i>i</i></sub> +  
<i>x</i><sup>2</sup>
```