ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (011) 818-5583 Fax (011) 818-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Continuação da segunda avaliação de linguagens e compiladores

Felipe Giunte Yoshida N° USP 4978231
Mariana Ramos Franco N° USP 5179364

4 de Dezembro de 2009

Construa o sistema de programação para a linguagem LazyComb que terá um compilador para a linguagem C, um ambiente de execução que contará com bibliotecas da linguagem. Não há E/S na linguagem, mas cada resultado deve ser mostrado na forma de combinadores.

1 Léxico

Para implementar o analisador léxico utilizamos a mesma estrutura criada para o Compilador FM, sendo necessário apenas modificar o arquivo **automato.xml** com a nova descrição do autômato mostrada a seguir:

```
<gramatica>
        <estado>
                <id>0</id>
                <final>false</final>
                <tipo>0</tipo>
                <transicao>
                        <entradas>ISK()</entradas>
                        <proximo>1</proximo>
                </transicao>
        </estado>
        <estado>
                <id>1</id>
                <final>true</final>
                <tipo>1</tipo>
        </estado>
</gramatica>
```

Como pode-se notar, o léxico aceita somente as entradas 'I','S', 'K', '(' e ')'. Assim, nosso compilador não trata entradas no formato Unlambda, Iota ou Jot.

2 Sintático

Assim como o léxico, também utilizamos a estrutura montada para o compilador FM para implementar o sintático. Foi necessário alterar a o arquivo **gramática.txt** de entrada para o meta-analisador com a gramática da linguagem Lazy-K na notação de Wirth:

```
Program = { Expr } .

Expr = "I" | "K" | "S" | "(" { Expr } ")" .
```

Desta maneira obtemos o autômato de pilha estruturado da linguagem como mostrado nas figuras 1 e 2.

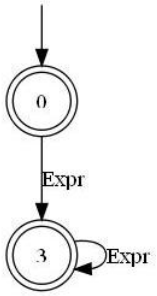

Figura 1: Submáquina 'Program'

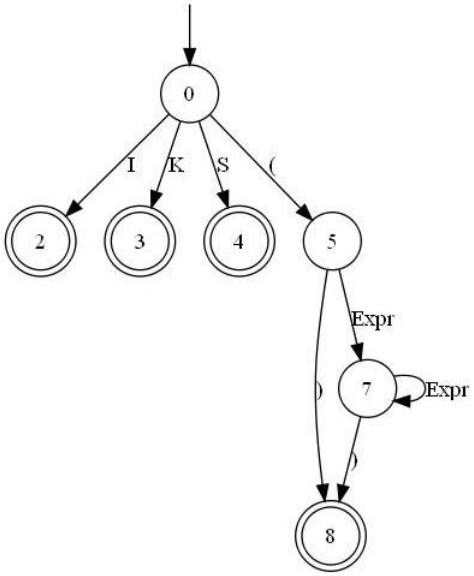

Figura 2: Submáquina 'Expr'

Também alteramos a classe **Sintatico.java** para que esta passasse a chamar corretamente os autômatos 'Program' ou 'Expr' conforme os tokens recebidos e o valor no topo da pilha estado-submáquina.

3 Semântico

Para fazer a análise semântica e gerar o código em C, responsável por reduzir a cadeia contida no arquivo **arquivofonte.lazy**, criamos uma nova classe **Semantico.java** que é chamada pelo sintático à cada transição nos autômatos 'Program' ou 'Expr'.

Basicamente o que o semântico faz é escrever no arquivo final as seguintes chamadas de função conforme o valor do token consumido:

```
if(token.getValor().equals("S")){
        main = "entra('S');";
        main = "tenta_reduzir();";

}else if(token.getValor().equals("K")){
        main = "entra('K');";
        main = "tenta_reduzir();";

}else if(token.getValor().equals("I")){
        main = "entra('I');";
        main = "tenta_reduzir();";

}else if(token.getValor().equals("(")){
        main = "novo_escopo();";


}else if(token.getValor().equals(")")){
        main = "fecha_escopo();";
        main = "tenta_reduzir();";
```

4 Ambiente de Execução

O ambiente de execução consiste em 5 funções, descritas abaixo:

novo_escopo Abre um novo escopo, que ocorre quando se encontra uma abertura de parênteses.

entra Insere um novo caracter no atual escopo

tenta_reduzir Tenta fazer uma das reduções ($Ix \rightarrow x$, $Kxy \rightarrow x$ ou $Sxyz \rightarrow (xz)(xy)$) no escopo atual

fecha_escopo Fecha escopo atual e copia resultado para o escopo anterior

imprime Imprime cadeia do escopo atual

5 Testes

Para a entrada `I(KKI)(IIKSI)`, chegamos a um cŮdigo em C, que nŃo serĞ colocado no relatŮrio por seu tamanho.

Ao se executar este cŮdigo em um compilador C, chegamos ao seguinte resultado:

```
Caractere adicionado I
I
Novo escopo nivel 1

Caractere adicionado K
K
Caractere adicionado K
KK
Caractere adicionado I
KKI
Reduziu K
K
Escopo nivel 1 fechado
IK
Reduziu I
K
Novo escopo nivel 1

Caractere adicionado I
I
Caractere adicionado I
II
Reduziu I
I
Caractere adicionado K
IK
Reduziu I
K
Caractere adicionado S
KS
Caractere adicionado I
KSI
Reduziu K
S
Escopo nivel 1 fechado
KS
KS
```