

Projeto Final INF1022

Alunos : Suemy Inagaki, Rafael Araujo Serpa

Matriculas: 1811208, 1520962



Introdução:

O trabalho consistiu em analisar a linguagem dada ProvolOne e sua gramática também fornecida. Após uma análise de ambas, deveríamos modificá-las para que perdessem as ambiguidades e assim pudessem ser devidamente compiladas por yacc. Após isso, o objetivo era escrever códigos .y e .l que fossem capazes de reconhecer uma entrada na linguagem ProvolOne modificada e conseguissem, a partir dela, escrever em alguma saída um código objeto em outra linguagem que pudesse ser propriamente executada.

Desenvolvimento:

Dado o que foi enunciado na Introdução, escolhemos como linguagem de saída C. Também como foi dito anteriormente, fizemos modificações na linguagem ProvolOne e em sua gramática para que ficasse de acordo com as exigências do problema. Para redigir nossos códigos, usamos o exemplo 2 apresentado no site da disciplina como modelo. Modificamos o arquivo a partir dele para chegar no resultado.

- Primeiro, é possível notar que as execuções “SE ID ENTAO cmd” e “SE ID ENTAO cmd SENAO cmd” geram ambiguidade. Por ex, a palavra “SE ID ENTAO SE ID ENTAO cmd SENAO cmd” tem duas formas de derivação. é possível começar com SE ID ENTAO cmd e ai trocar cmd por SE ID ENTAO cmd SENAO cmd ou então começar com SE ID ENTAO cmd SENAO cmd e então trocar o cmd do meio por SE ID ENTAO, gerando assim duas formas de derivação.
Há diversas formas de consertar esse problema, mas a solução que adotamos foi acrescentar um FIM ao final de cada uma das expressões, possibilitando assim que elas se distingam naturalmente.
- Seguindo o modelo do sequência comandos, para melhorar o reconhecimento da linguagem no terminal, colocamos o que seria equivalente ao “ZEILENENDE”, que chamamos de FIML.
- Como colocamos um FIM ao final dos comandos “if’s”, trocamos o comando FIM inicial por END
- Como queríamos realizar a mudança para C, faz sentido pensarmos em um único retorno da linguagem, então alteramos o escopo inicial de:” ENTRADA varlist SAIDA varlist cmds FIM” para “ENTRADA varlist SAIDA ret cmds END FIML”. Dessa forma, varlist pode ser diversas variáveis(múltiplas entradas), mas ret pode ser apenas uma(uma saída).

O procedimento foi: Abrimos um arquivo de saída, e conforme fomos lendo a entrada na linha de comando, fomos traduzindo para a linguagem C e alocando em vetores. Ao final, colocamos os vetores no arquivo de saída, que terá a sequência de comandos ProvolOne em C.

Exemplos:

1) ENTRADA A, B SAIDA B A=B END

Codigo objeto

int A; int B;

A = B;

return B;

2) ENTRADA A, B SAIDA B INC(A) END

Codigo objeto

int A; int B;

A++;

return B;

3) ENTRADA A SAIDA B ZERA(A) END

Codigo objeto

int A;

A = 0;

return B;

4) ENTRADA A, B, C SAIDA D SE F ENTAO INC(G) FIM ZERA(A) END

Codigo objeto

int A; int B;

int C;

if (F) {

G++;

}

```
A = 0;
```

```
return D;
```

5) ENTRADA A SAIDA B SE A ENTAO ZERA(A) SENAO INC(A) FIM END

Codigo objeto

```
int A;  
if (A) {  
    A = 0;  
} else {  
    A++;  
}
```

```
return B;
```

6) ENTRADA A, B, C SAIDA D SE B ENTAO A=B INC(C) ZERA(D) FIM END

Codigo objeto

```
int A; int B;  
int C;
```

```
if (B) {  
    A = B;  
    C++;
```

```
D = 0;
```

```
}
```

```
return D;
```

7) ENTRADA A, B, C SAIDA D SE B ENTAO INC(B) SE B ENTAO INC(B) FIM
FIM END

Codigo objeto

```
int A; int B;
```

```
int C;  
if (B) {  
    B++;  
    if (B) {  
        B++;  
    }  
}  
return D;
```