

UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS E TECNOLOGIAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

COMPILADORES

TRABALHO FINAL DE COMPILADORES

Milton Pedro Pagliusi Neto
Eduardo Pandini

Joinville

2022

1 Relatório

Neste trabalho da matéria de Compiladores, tínhamos como objetivo o desenvolvimento de um analisador sintático e o suficiente para a geração de um código compilável.

1.1 Geração de código intermediário

- start: Chamada da criação de cabeçalho.
- logico_if: Condição lógica "if", onde caso uma condição lógica seja verdadeira, a função dentro do "if" é realizada
- when: Condição lógica "when"
- declaração: Declaração de tipos, sendo estes:
 - Tipo com identificador. (int n)
 - Tipo com identificador e atribuição de valor. (int n = 1)
- declaracao_funcao: Estrutura de declaração de uma função simples, lendo seu tipo, escopo e argumentos.
- comentario: Estrutura responsável por reconhecer quando um comentário é inserido no código.
- chamada_funcao: Estrutura de chamada de uma função determinada.
- incremento: Estrutura de função de operação matemática.
- loop_while: Laço de repetição "while"
- loop_for: Laço de repetição "for"
- loop_do: Precursor do que seria uma chamada de função, na sua chamada "do" executa os comandos em seguida.
- print: Reconhece a palavra reservada "print" e mostra as informações desejadas no terminal
- assign: Atribui um novo valor a uma variável previamente declarada
- expression: Estrutura de reconhecimento de uma expressão "n" no código, assim como realizar operações matemáticas.

- `primitive_type`: Estrutura de reconhecimento de tipos primitivos no código.
- `b_expression`: Estrutura de reconhecimento de uma expressão binária/booleana.

1.2 Implementação dos comandos

- declaração de variável: as variáveis tem seu tipo declarado pelo usuário no momento da sua definição, portanto para uma declaração, primeiramente é definido o tipo, seguido do nome da variável, e opcionalmente um valor inicial, para este caso é necessário o sinal "=" seguido de um valor compatível com o tipo da variável.
- atribuição de valor: uma variável já declarada pode ter seu valor modificado, seja por atribuição manual ou por uso de expressões matemáticas, primeiramente informando o nome da variável já declarada a qual deseja se modificar o valor, seguido do sinal "=" e o novo valor ou expressão matemática utilizada para gerar o novo valor.
- Laço de repetição:

`while`: Laço de repetição que realiza o bloco de código declarado dentro de si enquanto uma condição é satisfeita, invocado utilizando a palavra "while", seguida de parênteses que contêm a condição a ser comparada para a realização do bloco de código, fechamento dos parênteses, abre chaves para indicar o código que será realizado caso a condição seja verdadeira, o código a ser realizado, e finalizado com o fechamento das chaves.

`for`: Laço de repetição que realiza o bloco de código declarado dentro de si enquanto uma condição é satisfeita, e ao mesmo tempo opera sobre a condição durante cada iteração. invocado utilizando a palavra "for", seguida da abertura de parênteses, o nome da variável que irá ser utilizada como parâmetro, ponto e vírgula, a condição a ser utilizada para verificar se a operação será realizada, ponto e vírgula, e a operação a ser feita na variável ao final de cada iteração sobre o laço, é feito então o fechamento dos parênteses e abertura de chaves para identificar o bloco de comandos a serem realizados, e por fim o fechamento das chaves.

- Condição lógica IF: uma estrutura que, caso tenha sua condição lógica satisfeita, opera sobre um bloco de código, podendo ou não possuir uma condição para caso a condição não seja satisfeita. invocada utilizando a palavra "if", abertura de parênteses, a condição lógica a ser avaliada, fechamento de parênteses, abertura de chaves para identificar o bloco de código a ser realizado caso a condição seja satisfeita, encerrado pelo fechamento de chaves, opcionalmente o fechamento de chaves pode ser seguido pela palavra "else" e outra abertura de chaves, contendo outro bloco de código finalizado pelo fechamento das chaves, este será o código que será executado caso a condição lógica não seja atendida.

1.3 Dificuldades encontradas

Tivemos dificuldades no reconhecimento e manuseio de diferentes tipos primitivos como BOOL, FLOAT ou até mesmo no caso do vetor.

Também foi encontrado como uma tarefa difícil uma declaração de uma chamada de função dentro do código, não conseguimos pensar em um escopo que conseguisse fazer com que a lógica do parser e do Jasmin "chamar" uma função.

Uma atribuição em sequência foi problemática ao tentarmos implementar no bytecode.

1.4 Exemplos de códigos

Listing 1.1 – Sequencia de fibonnaci:

```
int i;  
int fib1 = 1;  
int fib2 = 1;  
int soma;  
print("Sequencia de fibonacci");  
for(i = 1; i <= 10; i++) {  
    soma = fib1 + fib2;  
    fib1 = fib2;  
    fib2 = soma;  
    print(soma);  
}
```

Listing 1.2 – Caso teste site modificado:

```
// Exemplo de uso da linguagem com 0002-2022-01
```

```
int valor1 = 0;  
int valor2 = 0;  
  
int a;  
int g;  
int i;  
a = 5;  
  
print("Digite um numero:\n");  
print("5\n");  
while(i < 10) {  
    i++;  
}
```

```
        a++;
    }

    print ( "Incrementado algumas vezes a fica " );

    for (g=1;g < 10;g++){
        print ( ". " );
    }
    print ( "\n" );
    if (a==15){
        print ( "A=15\n" );
    }
    if ( a==18 or a==20 ){
        print ( "A igual a 18 ou 20\n" );
    }

    if (i==100){
        print ( "Decorando o texto ... \n" );
    }
    else{
        print ( "OK!\n" );
    }
}
```

1.5 Execução do programa

Para a execução do trabalho, os seguintes comandos devem ser executados no terminal:

```
make run
make java_run
```

1.6 Ambiente

Esse trabalho foi compilado utilizando um processador AMD Ryzen 5 3600 @ 3.6GHz, utilizando a Máquina Virtual disponibilizada pelo Prof. Dr. Ricardo Ferreira Martins.

1.7 Considerações finais

Acreditamos que com o prazo curto que tivemos nesse semestre de 2022/1, o desenvolvimento foi satisfatório suficientemente ao conseguir interpretar tipos primitivos

assim como assimilar valores, laços de repetição.