

Descrição do trabalho

O objetivo deste trabalho é a implementação de um compilador completo para MIPS a partir de uma linguagem bastante simples, descrita abaixo.

A saída do compilador deve ser programa em MIPS que ao ser executado (ou simulado) deve retornar o valor retornado pela 1ª função.

O programa gerado deve ser capaz de ser executado no ambiente MARS e ao final da execução deve ser mostrado na tela o resultado.

A implementação pode ser feita em Java, C++ ou Python*.

O trabalho pode ser feito individualmente, em duplas ou triplas.

Deve ser entregue uma descrição do projeto, indicando a contribuição de cada componente da equipe, além dos códigos-fonte e executáveis do projeto.

Prazo: 5 de julho às 23:55

Especificação Léxica

- Espaços em branco: [\n\t\r\f]
- Comentários: dois tipos de comentário, um começando com // e indo até o final da linha, o outro começando com /* e terminando com */, sem aninhamento
- Palavras reservadas: if, then, else, def
- Identificadores (id): uma letra, seguido de zero ou mais letras, dígitos ou _
- Numerais (int): apenas números inteiros
- Operadores relacionais (opRel): >, <, =
- Operadores aritméticos (opArit): +, -, *, /
- Sinais de pontuação: (,), ;, ,

Sintaxe

A sintaxe é dada usando EBNF. Meta-símbolos usados como tokens estão entre aspas simples. Os operadores aritméticos seguem a precedência habitual, assim como a avaliação de IF's aninhados.

```
P → id '=' int ; P | I
I → D ';' I | D
D → def id '(' 'ARGS' ')' '=' E ';'
ARGS → id ',' ' ' ARGS | id
SEQ → E | SEQ ',' E
E → int | id | if E opRel E then E else E
    | E opArit E | id '(' SEQ ')'
```

Semântica

A análise semântica deve verificar se:

- a) os identificadores que aparecem em cada função são passados como parâmetros,
- b) as funções chamadas foram definidas e com a quantidade de parâmetros compatível,
- c) não há parâmetros repetidos (na declaração de uma função),
- d) não há funções declaradas mais de uma vez.

Exemplo

```
a = 312;
b = 111;
def mdc(a,b) =
    if mod(a,b) = 0
    then b
    else mdc(b, mod(a,b));
def mod(a,b) =
    if a < b
    then a
    else mod(a-b,b)
```

*Caso deseje usar outra linguagem além das referidas, entre contato com o professor.