

Primeira Prova – valor 20 pontos (4-6-10)

Considere uma linguagem imperativa, onde os programas têm a seguinte forma geral:

Program id; declarações begin comandos end.

onde **Program** é uma palavra reservada, **id** é o identificador do programa e **comandos** uma série de zero ou mais comandos separados por ponto-e-vírgula. Um programa pode possuir zero ou mais seções de declaração de variáveis que são da forma: **Var lista;** sendo *lista* uma sequência de uma ou mais declarações da forma: **lista_ids : tipo;** onde *lista_ids* é uma sequência de um ou mais identificadores separados por vírgulas. Os tipos básicos são o inteiro e o real, representados pelas palavras reservadas **integer** e **real**. Variáveis do tipo apontadores são declaradas com o símbolo ^ antes do tipo básico. Vários níveis de apontadores são permitidos. Números podem ser inteiros ou reais, os quais podem não ter parte fracionária mas têm parte inteira obrigatória. Os identificadores são compostos de letras, dígitos e sublinhado e não podem começar com dígito. Comentários são limitados por { }. Maiúsculas e minúsculas NÃO são diferenciadas.

Os comandos permitidos são:

- Comando de atribuição, da forma: **objeto := expressão** onde *objeto* pode ser um identificador ou o conteúdo apontado por um identificador. Neste caso, o identificador é seguido por ^ (um ou mais).
- Comandos de teste, da forma **if expressão then comando1 else comando2**, onde a parte do senão é opcional. *Comando1* e *2* podem ser substituídos por blocos da forma **begin comandos end** onde *comandos* são uma lista de zero ou mais comandos separados por ponto-e-vírgula.
- Comandos de repetição, da forma **for id:=expressão to expressão do comando**. *Comando* pode ser substituído por um bloco, como descrito para o comando **if**

Expressões misturam variáveis, conteúdos apontados por identificadores e números, através de operadores de adição (+), subtração (-), multiplicação (*) e divisão (/), além de parênteses, observando-se as regras básicas de precedência. Expressões relacionais para teste são da forma **exp1 relacional exp2**, onde *exp1* e *exp2* são expressões e *relacional* pode ser >, >= ou =.

Um exemplo de programa Pascal é dado a seguir:

```
Program Exemplo;
  Var x,y:integer; z:real;          {declarações}
  Var p:^integer;                  {apontador de apontador de inteiro}
{bloco principal de comandos}
Begin
  x:=2*(3-1);
  p^:=x+1;
  for y:=0 to x+1 do begin
    if y=1 then z:=5. else begin z:=1; x:=0; end;
    p^:=p^+1;
  end;
end.
```

- Determine o alfabeto da linguagem e o padrão de formação dos lexemas. No caso das palavras reservadas formadas por letras, basta fornecer o padrão de formação de uma delas que tenha mais de 5 letras.
- Escreva um autômato finito determinístico que represente o analisador léxico para a linguagem descrita. Indique os casos onde há necessidade de se devolver o último caractere lido.
- Escreva uma gramática LL(1) para a linguagem descrita, observando-se a precedência de operadores nas expressões.