Analisador Léxico

O objetivo do trabalho é construir um Analisador Léxico para realizar a análise léxica de um programa escrito em uma linguagem baseada Pascal (Portugol), mas com identificadores em português. O Analisador Léxico deve ler um o programa fonte em um arquivo texto e identificar os átomos (tokens) da linguagem. O arquivo de entrada deve ser fornecido para o programa através da **linha de comando**.

No compilador a análise léxica é realiza como uma sub-rotina do Analisador Sintático, sendo assim, para que seja possível a reutilização do Analisador Léxico pelo Analisador Sintático, deverá ser implementada uma rotina responsável na pela análise léxica (`AnaLex()`), que posteriormente na fase de Análise Sintática está rotina será a interface entre as duas análises.

Para cada átomo encontrado, a rotina `AnaLex()` retorna uma estrutura com informações referentes ao átomo contendo além do átomo encontrado, a linha que gerou o átomo e o seu atributo (caso seja necessário). Além disso o analisador léxico faz um controle das linhas do programa fonte e também a eliminação dos delimitadores (espaços em branco, tabulação, nova linha e retorno de carro). Para cada átomo reconhecido devem ser apresentados os seguintes valores:

{*Número da Linha do Átomo, Átomo, Atributo*}

Caso ocorra um erro no reconhecimento de um dos átomos do programa fonte analisado, o seu analisador léxico deve parar a execução com uma mensagem informando a linha onde ocorreu o **erro**. Nas seções a seguir são apresentadas as definições regulares para os átomos que a rotina AnaLex deve identificar.

1. Definição dos Identificadores da Linguagem

`LETRA` → `[A-Za-z]`
`DIGITO` → `[0-9]`
IDENTIFICADOR → LETRA (LETRA | DIGITO| _)*

As palavras reservadas em uma linguagem de programação não podem ser utilizadas como identificadores, para simplificar a rotina de análise léxica, o reconhecimento das palavras reservadas deverá ser feita com base na definição regular de identificadores.

Quando um **IDENTIFICADOR** for reconhecido, deve-se realizar uma **busca binária** na tabela de palavras reservadas; se for encontrado, deve-se retornar o átomo associado com a palavra reservada encontrada; caso contrário o átomo **IDENTIFICADOR** deve ser retornado com o seu atributo, a seqüência de caracteres que gerou o átomo.

Importante:
O tamanho do identificador deve ser limitado em 16 caracteres, caso seja encontrado um identificador maior que o limite o AnaLex devolverá um erro. Além disso, conforme a expressão regular acima um identificador pode ter depois da primeira LETRA o caractere *underline* "_".

A seguir é apresentada a lista dos átomos que devem ser retornados quando uma das palavras reservadas for reconhecida pela rotina de análise léxica.

Átomos Retornados	Descrição
ALGORITMO	Inicio do programa
CARACTERE	Tipo de dado para definir um caractere
DIV	Operador de divisão
E	E lógico
ENQUANTO	Determina um laço com condição no início
ENTAO	Usado no bloco de estrutura de seleção
FACA	Usado no bloco de estrutura de repetição
FALSO	Constante booleana
FIM	Fim de um bloco ou do programa
FUNCAO	Determina uma função no algoritmo
INICIO	Início de uma estrutura
INTEIRO	Tipo de dado para números inteiros
LOGICO	Tipo booleano
MOD	Operador de Resto
NADA	Usado para métodos sem parâmetros
NAO	Negação lógica
OU	OU lógico
PROCEDIMENTO	Determina um procedimento no algoritmo
REAL	Tipo de Dado para números reais (ponto flutuante)
REF	Operador para de referência para variáveis
SE	Determina uma estrutura de seleção
SENAO	Caso contrário da instrução de seleção
VARIAVEIS	Início do bloco de declaração de variáveis
VETOR	Declara uma estrutura seqüencial unidimensional

O AnaLex não é sensível ao caso, ou seja, as Palavras Reservadas e os IDENTIFICADORES podem ser informadas com caracteres maiúsculos ou minúsculos, sendo assim, para o lexema "**alGoRiTmo**" ou "ALGORITMO" o átomo retornado é **ALGORITMO**.

2. Definição dos átomos simples

ABRE_PAR → (
ATRIBUICAO → :=
FECHA_PAR →)
PONTO_VIRGULA → ;
PONTO_PONTO → ..
ABRE_COLCHETES → [
FECHA_COLCHETES →]

Toda vez que for reconhecido um dos símbolos a rotina AnaLex retorna o átomo correspondente a definição regular do símbolo

3. Operadores Aritméticos

SUBTRACAO → **-**
ADICAO → +
MULTIPLICACAO → *

Toda vez que for reconhecido um dos símbolos a rotina AnaLex retorna o átomo correspondente ao operador aritmético.

4. Operadores Relacionais (OP_RELACIONAL)

ME → <
MEI → <=
IG → =
DI → #
MA → >
MAI → >=

OP_RELACIONAL → ME | MEI | IG | DI | MA | MAI

Toda vez que for reconhecido um dos símbolos acima a rotina AnaLex retorna o átomo **OP_RELACIONAL** com seu atributo. O atributo associado ao átomo **OP_RELACIONAL** deverá ser as constantes simbólicas (`ME, MEI, IG, DI, MA, MAI`).

5. Comentários na Linguagem

```
NOVA_LINHA → {caracter de nova linha(=13) ou retorno de carro(=10) }
CHAR → {qualquer caractere ASCII } - NOVA_LINHA
COMENTARIO_1 → (* (CHAR | NOVA_LINHA)* *)
COMENTARIO_2 → { CHAR* }

COMENTARIO → COMENTARIO_1 | COMENTARIO_2
```

Para **COMENTARIO** é retornado átomo correspondente e o controle de linhas é mantido dentro do comentário. **COMENTARIO** não possui atributo.

6. Constantes

6.1. Constantes com valores inteiros e Reais

$OP\_EXP\_1 \rightarrow \hat{}(+|\lambda)DIGITO^{+}$

CONSTANTE_INTEIRO $\rightarrow DIGITOS^{+}(OP\_EXP\_1|\lambda)$

$OP\_FRACAO \rightarrow ,(DIGITOS+)$
$OP\_EXP\_2 \rightarrow \hat{}(+|-|\lambda)DIGITO^{+}$
CONSTANTE_REAL $\rightarrow DIGITOS^{+}OP\_FRACAO(OP\_EXP\_2|\lambda)$

A rotina `AnaLex()` retorna o átomo referente a constante encontrada CONSTANTE_INTEIRO ou CONSTANTE_REAL e o seu valor numérico como atributo. Lembrando que o símbolo λ representa a palavra vazia

6.2. Constante Frase na Linguagem

CONSTANTE_FRASE → `"CHAR*"`
A rotina `AnaLex()` retorna o átomo CONSTANTE_FRASE e a string que gerou o átomo como atributo. Considere que o tamanho da constante frase é limitada em 32 caracteres. A CONSTANTE_FRASE deve sempre ser finalizada com aspas na mesma linha onde foi iniciada sua declaração, ou seja, não temos quebra de linha em uma CONSTANTE_FRASE.

6.3. Constante caractere

CONSTANTE_CARACTERE → `'CHAR'`

A rotina `AnaLex()` retorna o átomo CONSTANTE_CARACTERE e caractere que gerou o átomo como atributo.

O trabalho deve ser implementado na linguagem C/C++ e deve ser bem documentado. A entrega do trabalho deve ser feita pelo *blackboard*, você deve enviar o programa fonte e os seus arquivos de testes.

No desenvolvimento do seu trabalho siga as orientações descritas nos sites:
http://www.ime.usp.br/~pf/algoritmos/aulas/layout.html
http://www.ime.usp.br/~pf/algoritmos/aulas/docu.html

Este trabalho pode ser feito **em dupla**, evidentemente você pode "discutir" o problema dado com seus colegas, inclusive as "dicas" para chegar às soluções, mas a dupla deve ser responsável pela solução final e pelo desenvolvimento do seu programa.