

Trabalho Prático I – Análise Léxica

Descrição do trabalho

Nesta etapa, você deverá implementar um analisador léxico para a linguagem **PortuGolo** cuja descrição encontra-se nas páginas 3 e 4.

Seu analisador léxico deverá ser implementado conforme visto em sala de aula, com o auxílio de um Autômato Finito Determinístico. Ele deverá reconhecer um lexema e retornar, a cada chamada, um token de acordo com o lexema encontrado.

Para facilitar a implementação, uma Tabela de Símbolos (TS) deverá ser usada. Essa tabela conterá, inicialmente, **todas as palavras reservadas** da linguagem. À medida que novos tokens forem sendo reconhecidos, esses deverão ser consultados na TS antes de serem cadastrados. Somente **palavras reservadas e identificadores serão cadastrados na TS**. Não é permitido o cadastro de um mesmo token mais de uma vez na TS.

Seu Analisador Léxico deverá imprimir a lista de todos os tokens reconhecidos, assim como mostrar o que está cadastrado na Tabela de Símbolos. Na impressão dos tokens, deverá aparecer a tupla <nome_do_token, lexema> assim como linha e coluna do token.

Além de reconhecer os tokens da linguagem, seu analisador léxico deverá detectar possíveis erros e reportá-los ao usuário. O programa deverá informar o erro e o local onde ocorreu (linha e coluna). Os erros verificados serão: *i*) caracteres desconhecidos (não esperados ou inválidos), *ii*) string não fechada antes de quebra de linha e *iii*) comentário (ou string) não-fechado antes do fim de arquivo.

Espaços em branco, tabulações, quebras de linhas e comentários não são tokens, ou seja, devem ser descartados/ignorados pelo referido analisador.

Na gramática do **PortuGolo**, os terminais de um lexema, bem como as palavras reservadas, estão entre aspas duplas para destacá-los, logo, **as aspas não são tokens**.

Cronograma e Valor

O trabalho vale 30 pontos no total. Ele deverá ser entregue por etapas, sendo a primeira etapa correspondendo ao Analisador Léxico, conforme consta na tabela abaixo.

Etapas	Data de entrega	Valor	Multa por atraso
Analisador Léxico	23/09/2017	10 pontos	2pts/dia
Analisador Sintático	A definir	10 pontos	2pts/dia
Analisador Semântico	A definir	10 pontos	2pts/dia

O que entregar?

Você deverá entregar nesta etapa:

1. Uma figura apresentando o Autômato Finito Determinístico para reconhecimento dos tokens, conforme visto em sala de aula (dê uma olhada na ferramenta JFLAP: <http://www.jflap.org/>);
2. Todos os arquivos fonte;
3. Relatório técnico contendo explicações do propósito de todas as classes, métodos ou funções da implementação;
4. O trabalho deverá ser entregue por e-mail para gustavo.fernandes@prof.unibh.br com o assunto TP1_COMP_CC2018_2.

Regras:

O trabalho poderá ser realizado individualmente ou em dupla.

Não é permitido o uso de ferramentas para geração do analisador léxico.

A implementação deverá ser realizada em uma das linguagens C, C++, Java, Ruby ou Python.

A recuperação de erro deverá ser em Modo Pânico, conforme discutido em sala. Mensagens de erros devem ser apresentadas indicando a linha e coluna de ocorrência do erro.

Trabalhos total ou parcialmente iguais receberão avaliação nula.

Ultrapassados cinco (5) dias, após a data definida para entrega, nenhum trabalho será recebido.

Em anexo (pasta: *lexer_exemplo*) segue um exemplo de uma Gramática, AFD e um lexer programado. **ATENÇÃO:** a gramática do exemplo não tem relação com a gramática do *PortuGolo*.

Gramática da linguagem *PortuGolo*

Compilador	→ Programa EOF
Programa	→ "algoritmo" ("declare" (DeclaraVar) ⁺)? ListaCmd "fim" "algoritmo" ListaRotina
DeclaraVar	→ ListaID Tipo ";"
ListaRotina	→ ListaRotina Rotina ε
Rotina	→ "subrotina" ID "(" ListaParam ")" ("declare" (DeclaraVar) ⁺)? ListaCmd Retorno "fim" "subrotina"
ListaParam	→ Param "," ListaParam Param
Param	→ ListaID Tipo
ListaID	→ ID "," ListaID ID
Retorno	→ "retorne" Expressao ε
Tipo	→ "logico" "numerico" "literal" "nulo"
ListaCmd	→ ListaCmd Cmd ε
Cmd	→ CmdSe CmdEnquanto CmdPara CmdRepita CmdAtrib CmdChamaRotina CmdEscreva CmdLeia
CmdSe	→ "se" "(" Expressao ")" "inicio" ListaCmd "fim" "se" "(" Expressao ")" "inicio" ListaCmd "fim" "senao" "inicio" ListaCmd "fim"
CmdEnquanto	→ "enquanto" "(" Expressao ")" "faca" "inicio" ListaCmd "fim"
ComandoPara	→ "para" CmdAtrib "ate" Expressao "faca" "inicio" ListaCmd "fim"
ComandoRepita	→ "repita" ListaCmd "ate" Expressao
CmdAtrib	→ ID "<--" Expressao ";"
CmdChamaRotina	→ ID "(" (Expressao ("," Expressao) [*])? ")" ";"
CmdEscreva	→ "escreva" "(" Expressao ")" ";"
CmdLeia	→ "leia" "(" ID ")" ";"
Expressao	→ Expressao Op Expressao ID ID "(" (Expressao ("," Expressao) [*])? ")" Numerico Literal "verdadeiro" "falso" OpUnario Expressao "(" Expressao ")"
Op	→ "Ou" "E" "<" "<=" ">" ">=" "=" "<>" "/" "*" "-" "+"
OpUnario	→ "Nao"

Padrões para números, literais e identificadores do *PortuGolo*

digito	= [0-9]
letra	= [A-Z a-z]
Numerico	= digito ⁺ (.digito ⁺)?
ID	= letra (letra digito) [*]
Literal	= pelo menos um dos 256 caracteres do conjunto ASCII entre aspas duplas

Outras características de *PortuGolo*

- As palavras-chave de *PortuGolo* são reservadas;
- A linguagem possui comentários de mais de uma linha. Um comentário começa com “/*” e deve terminar com “*/”;
- A linguagem possui comentários de uma linha. Um comentário começa com “//”;
- A linguagem não é *case-sensitive*;
- Cada tabulação, deverá contar como 3 espaços em branco;