Trabalho Prático I - 2018.2

1. Instalação do ANTLR e java:

- 1.1. É necessário instalar java. Pode ser obtido em https://www.java.com/
- 1.2. Para baixar o ANTLR deve-se executar as seguintes linhas de comando no terminal do Ubuntu (ou usar o arquivo jar presente na pasta Codigo do projeto):
  cd /usr/local/lib
  sudo wget https://www.antlr.org/download/antlr-4.7.2-complete.jar

2. Pré-execução

- 2.1. A cada vez que for aberto uma nova aba no terminal é preciso que sejam executados dois comandos:
  export CLASSPATH=".:/usr/local/lib/antlr-4.7.2-complete.jar:$CLASSPATH"
  alias grun='java org.antlr.v4.gui.TestRig'

3. Geração e execução do analisador léxico

- 3.1. Pelo terminal, entre na pasta Código (do projeto)
- 3.2. Execute o comando make. Para isso, use make FILE=arquivo.xpp, onde arquivo.xpp é o arquivo a ser analisado

4. Entendendo o que está acontecendo

- 4.1. Com base no arquivo Xcc.g4 o comando make invocará o ANTLR para gerar o analisador léxico e sintático
- 4.2. Em seguida o analisador léxico é invocado pela classe xcc.java, contida na pasta Codigo do projeto
- 4.3. A classe xcc.java salvará um arquivo ListaDeToken.txt contendo todas as ocorrências de tokens do código fonte, na ordem em que aparecem. Se um token aparecer mais de uma vez, todas as ocorrências dele serão salvas no arquivo
- 4.4. A classe xcc.java também salvará um arquivo TabelaDeSimbolos.txt
- 4.5. Os arquivos .class e .java gerados pelo ANTLR serão removidos (incluindo o parser, que não é para ser entregue no trabalho 1), permanecendo apenas os arquivos úteis

5. Entendendo a gramática do arquivo Xcc.g4

- 5.1. grammar Xcc deve ser declarado com o nome do arquivo. Se o nome do arquivo fosse oi.g4, então teria de ser declarado grammar oi
- 5.2. Apesar da gramática do parser não ser necessária para o analisador léxico, o ANTLR exige que seja informado pois o parser é gerado no mesmo comando do léxico
- 5.3. Após informar a gramática são declaradas as regras do analisador léxico no formato TOKEN: PADRAO
- 5.3.1. As palavras reservadas devem aparecer antes das demais
- 5.4. As regras declaradas como fragment servem apenas para formação de outras regras, portanto são desconsideradas no restante do processo de análise léxica. Ex: o fragment "DIGIT" é usado por "INTCONSTANT": "INTCONSTANT : (DIGIT)+;". Fora de "INTCONSTANT" o "DIGIT" é desconsiderado.
- 5.5. "[a-z]", "[A-Z]" e "[0-9]" correspondem a intervalos de caracteres ou digitos. No primeiro são considerados os caracteres de "a" a "z" minusculos, no segundo é o intervalo de caracteres maiusculos e o último corresponde aos dígitos de "0" a "9"
- 5.6. [\u0060] e [\u00C0-\u00C5] são intervalos semelhantes aos do item 5.5, porém correspondem a caracteres em unicode. Foram usados caracteres em unicode por conta da incompatibilidade da declaração explícita com o ANTLR.
- 5.6.1. Esse link mostra uma lista de caracteres em unicode