

Relatório de Implementação: Analisador Léxico de 2 passos

O programa utiliza uma abordagem simplificada para análise léxica utilizando um conjunto de 4 autômatos:

1. Um especializado para encontrar apenas palavras compostas unicamente por letras
2. Um exclusivo para números
3. Um especializado para símbolos matemáticos(+, -, --, ++, *, /)
4. Outro somente para símbolos lógicos(<, <=, >, >=, =, ==, !=, !=|, ||, &, &&, ^, ~)

Para tanto, é necessário ler o arquivo de entrada, e separar cada linha da entrada em substrings, onde cada substring é qualquer sequência de termos, separada por um espaço em branco ' ', ou um carácter de separação ('.', ';', '\t', '\n' ou '?').

Por conta disso, para facilitar o trabalho do parser e garantir o reconhecimento correto das sequências, todos os símbolos nos arquivos de entrada foram espaçados com, pelo menos, um espaço em branco entre eles.

Arquivo: Expr.rpp
expR = expA + expA - expB * expX / expF + 235
Arquivo: Expr2.rpp
for (int i = 0 ; i < (const) 2 ; i ++) { expR = expA + expA - expB * expX / expF }
Arquivo: Expr3.rpp
< > <= >= = == != ! ~ & && + ++ - -- * / ^