

Universidade do estado de Santa Catarina –UDESC- CCT

Disciplina: Compiladores

Professor: Ricardo Ferreira Martins

Alunos: Paulo Henrique Gonçalves, Brian Laus Bertemes

Trabalho 1

## **Criação de analisador léxico**

## 1. Informações iniciais:

O analisador léxico foi desenvolvido usando o software flex em sua versão 2.6.4 em kernel linux. O código, visando melhor manutenibilidade, foi dividido em 3 arquivos, o próprio analisador, no arquivo com extensão .lex, os headers da função usadas, no arquivo com extensão .h, e o código dessas funções, no arquivo .c.

Para a criação da tabela de símbolos foi escolhido o uso de uma lista duplamente encadeada, onde cada nodo é uma struct que armazena o token, o tipo do token, a linha, coluna, e um ID para melhor organização.

## 2. Mudanças na gramática:

Ao analisar o algoritmo passado para análise, percebemos a necessidade da criação de alguns tokens extras como for, return, print e read que são palavras que eram reconhecidas como palavras comuns, mas devem ser reservadas. Outros erros de sintaxe como “=” que era reconhecido como um operador lógico e não de atribuição, “FALSE” que estava em letra maiúscula enquanto o analisador só reconhece em letra minúscula, ou bool que na gramática deve ser escrito boolean, não foram feitas alterações no analisador para eles, e sim no código, uma vez que havia equivalentes para sua função em nossa gramática.

Foi adicionado também um token para reconhecimento de comentários “//” e de chave “{ }” para diferenciação de escopo de funções, bem como letras maiúsculas para reconhecimento em palavras e letras, além de adicionar o reconhecimento de aspas duplas em string, ignorando assim o que está dentro dela para não se preocupar com todas as letras do idioma escrito, mas salvando a string em token, e adicionamos também underline (\_). Decidimos adicionar esses tokens pois são usados frequentemente em códigos e não havia motivo para não os aceitar.