TopScript

Vinicios Henrique Wentz[1], Rafael Henrique Tibola[1], Felipe Divensi[1]

[1]Universidade Tecnologica Federal do Paraná (UTFPR)
Av. Brasil, 4232 - Independência, Medianeira - PR, 85884-000

{vinicioswentz, tibola, felipee}@alunos.utfpr.edu.br

Abstract. *aaaa*

Resumo. *aaa*

1. Compilador

Linguagens de programação são notações para a descrição de computações para pessoas e maquinas. Hoje, no mundo em que vivemos as pessoas são totalmente dependente das linguagens de programação, pois, todos os software os quais os computadores rodam são escritos em alguma linguagem de programação. Mas para que isso seja possível, precisa-se de alguma forma que os computadores entendam essas linguagens de programação, software que faz está tradução é chamado de compilador [Alfred V. Aho 2007].

Pode-se definir compiladores como programas que fazem a tradução de outros programas os quais foram escritos em alguma linguagem e os preparam para serem executados. Para fazer está transformação o compilador deve entender [K. Cooper 2012].

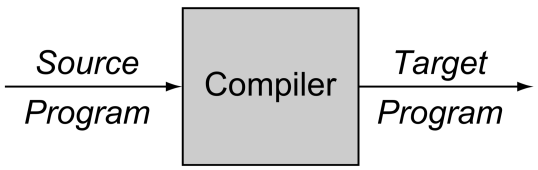

Figure 1. Compilador.

A estrutura de um compilador é compostas por duas partes a parte de analise e a de síntese. A fase de analise quebra o programa fonte em peças constituintes é impões uma estrutura gramatical. Essa estrutura é utilizada para a criação da representação intermediária do programa fonte. Caso a fase de analise detecte que o programa está sintaticamente mal formado ou não está semanticamente claro, essa fase deve prover informações informativas para que o usuário tome as devidas providencias. A analise também coleta algumas informações importante e as armazenas em uma estrutura chamada tabela de símbolo, a qual é passada com a representação intermediária para a analise sintática. A analise sintática é responsável pela construção do programa alvo desejado. Ela usa a representação intermediária e a tabela se símbolo para tal processo [Alfred V. Aho 2007]. Essas fases são chamadas de *front end* e *back end* respectivamente. A fase será abordada com maior detalhes nas Seções 2, 4 e 3 e a síntese na Seção.

2. Analise Léxica

A tarefa principal do analisador léxico na primeira fase da compilação é de ler o programa fonte, agrupa-los em lexemes e produzir uma sequencia de *tokens* para cada lexeme no programa fonte. Muitas vezes o analisador léxico interage com a tabela de simbolos. Quando o analisador léxico processa um lexeme que pertence a um identificador ele precisa adiciona-lo na tabela de simbolos [Alfred V. Aho 2007]. Essa iteração é demonstrada na Figura **??**.

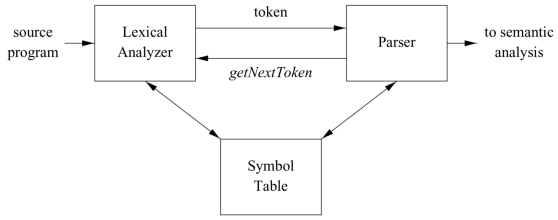

Figure 2. Iterações entre o analisador lexico e o *parser*.

3. Analise Sintática

4. Analise Semântica

5. Documentação

References

Alfred V. Aho, Monica S. Lam, R. S. J. D. U. (2007). *Compilers: principles, techniques, and tools*. Pearson/Addison Wesley, 2 edition.

K. Cooper, L. T. (2012). *Engineering a Compiler*. Morgan Kaufman, 2 edition.