NOME: RAFAELLE LARISA DE ARRUDA DATA: 07/10/2020

Prof: BRAULIO ADRIANO DE MELLO DISCIPLINA: Linguagens formais e autômatos

Atividade Avaliativa

1- Descrever, de forma objetiva, a finalidade de cada uma das seguintes etapas de compilação. Analise léxica, Analise sintática, Analise semântica, geração de codigo intermediario, e otimização.

* Analise léxica → É DE TODO conhecimento geral, que analise léxica quebra a entrada em palavras conhecidas como tokens (é um conjunto de caracteres), entretanto, recebe uma sequência de caracteres e produz uma sequencia de palavras chaves, pontuação e nomes, ainda descarta comentarios e espaços em branco.

* Analise sintática → analisa a estrutura de frases do programa, entretanto é a parte da gramatica que estuda a disposição das palavras na frase e das frases em um discurso, basicamente essa etapa no processo de compilação deve reconhecer as formas do programa fonte e determinar se ele é valido ou não, mas a principal tarefa é determinar se o programa de entrada representado pelo fluxo de tokens possui as sentenças validas para linguagem de programação.

* Analise Semântica → É responsavél por verificar aspectos relacionados ao siguificado das instruções, essa é a terceira etapa do processo de compilação e nesse momento ocorre a validação de uma serie regras que não podem ser verificadas nas etapas anteriores, é importante ressaltar que muitos dos erros semanticos tem origem de regras dependentes da linguagem de programação. As validações que não podem ser executadas pelas etapas anteriores devem ser executadas durante a analise semântica a fim de garantir que o programa fonte esteja coerente e o mesmo possa ser convertido para linguagem de maquina, ressaltamos ainda que a analise semântica percorre a arvore sintática relacionada os identificadores com seus dependentes de acordo com a estrutura hierarquia, um importante componente da analise é a verificação de tipos, nela o compilador verifica se cada operador recebe os operandos permitinde e especificados na linguagem fonte.

GERAÇÃO DE codigos intermediários → A representação intermediária é um programa para maquina abstrata, que pode ter varias formas, devendo ter dois aspectos importantes, ser fácil de produzir e fácil de traduzir no programa alvo, A linguagem utilizada para geração de um codigo em formato intermediário entre a linguagem de alto nivel e a linguagem assembly deve representar todas as expressões do programa original, de forma independente do processador para o qual o programa sera gerado. Concluindo que a geração de codigos intermediário é a transformação da Arvore de derivação em um segmento de codigo, esse codigo pode, eventualmente, ser o codigo objeto final, mas, na maioria das vezes, constitui-se num codigo intermediario.

otimização → A fase de otimização tenta melhorar o codigo intermediario, de forma que venha resultar um codigo de máquina mais rapido em tempo de execução. Existem otimizações das mais basicas até mais complexas. A representação intermediaria é gerada de forma que obtenha a mesma saida do codigo fonte, entanto, ás vezes, o codigo gerado pode ser muito mais eficiente através da eliminação de alguns comandos desnecessario. O objetivo da etapa de otimização de codigo é aplicar um conjunto de tecnicas para detectar tais sequencias e substitui-las por outras que removem as situações de ineficiência. Ressaltamos uma enorme quantidade de formas de otimizações de codigo que cada compilador executa, entretanto, os compiladores que realizam estas otimizações despendem grande parte para realiza-las. Concluimos que pode resolver problemas como, por exemplo, instruções invariantes em laços, rearranjamento os simbolos sintaticos antes de passá-los para analise semântica.

2- Para linguagens do tipo 3, 2, 1 e 0, quais são as respectivas máquinas reconhecedoras?

tipo 3 (Linguagem Regular) é mais adequado uso de automato finito.

tipo 2 (Gramatica livre de contexto) é mais adequado o uso de automato com pilha.

tipo 1 (Gramatica Sensivel ao contexto) é mais adequado o uso da Maquina de Turing com memoria limitada.

tipo 0 (Gramatica irrestrita), é mais adequado do uso da maquina de Turing.

3- Construa uma gramatica para a seguinte Linguagem:

$L(G) = \{ x \mid x \in (0,1)^+$ onde as cadeias nunca iniciam por $0\}$

Podemos responder da seguinte maneira:

$S := 1A$

$A := S|0|1|0A|\epsilon$