

REOblade: Student's Sacrifice (7)

Mateus Carvalho Gonçalves - 201810245

A Geração de Código Intermediário é a etapa que divide o front-end e o back-end da compilação. As análises (front-end) produzem como resultado uma árvore sintática que contém muitas informações desnecessárias para o processo de otimização e tradução para código objeto (back-end). Por isso, a representação em um padrão de CI tem como objetivo facilitar os processos de back-end e desacoplar as etapas de compilação, visto que a representação intermediária é independente de máquina, ou seja, não especifica detalhes da máquina alvo, como uso de registradores e referenciamento à endereços de memória.

Essa etapa não é obrigatória na compilação, porém um compilador que não implementa um padrão de CI precisa criar todo o back-end especificamente para aquela linguagem considerando separadamente cada arquitetura de processadores existente. Por isso, se tornou usual as representações em CI. Atualmente existem padrões amplamente estudados pela academia e adotados pela indústria, como o LLVM, que conseguem implementar inúmeras otimizações em tempo de GCI, e com essa adoção massiva, os fabricantes dos processadores começaram a criar rotinas que suportam esse padrão e facilitam/otimizam a compilação e execução ainda mais.

Uma das principais técnicas de GCI é a Tradução Dirigida pela Sintaxe, isso implica que ela pode ser executada paralelamente às análises sintática e semântica. Existem vários padrões para representação do CI, por exemplo, Árvore Sintática Abstrata (AST), forma posfixa e Código de Três Endereços (TAC).

- AST: Uma árvore sintática que esconde informações desnecessárias para o back-end, como símbolos não-terminais;
- Forma posfixa: notação que beneficia máquinas baseadas em pilhas e é caracterizada pela posição dos operadores após os operandos;
- TAC: é a forma mais popular para representações intermediárias.
 - É uma sequência de comandos, similares a produções sintáticas, que utilizam até 3 endereços por produção;
 - Exemplo: $A := B \text{ op } C$;
 - Também representa saltos em código;
 - Compilador cria variáveis temporárias para auxiliar no processo, parecidas com símbolos não terminais. Essas variáveis podem ser usadas tanto do lado direito quanto do lado esquerdo das produções;
 - $A = B + C * D \Rightarrow t1 = C * D$
 $t2 = B + t1$
 $A = t2$