OPTIMIZACION DE CODIGO

Introducción

Bajo el epigrafe "Optimización de código" se incluyen diversas técnicas tendentes a mejorar la calidad del código objeto generado por el compilador. Esta mejora puede hacerse desde el punto de vista de la utilización de memória o desde el punto de vista del tiempo de ejecución. Ambas perspectivas no son en principio incompatibles, si bien en determinados casos se plantea la necesidad de elegir entre ambas, o bien establecer un cierto compromiso. Actualmente el factor suele considerarse mas importante.

Aunque tradicionalmente se emplea el término "optimización", el objetivo que se pretende alcanzar no es el de conseguir el mejor de entre todos los posibles códigos equivalentes, sino simplemente mejorar mediante técnicas heuristicas el código que se obtiene mediante la traducción automática de las fases anteriores. La obtención de un código óptimo sería un problema mucho mas complejo y dificil de abordar. De hecho, antes de incluir en un compilador una estrategia de optimización determinada, debe ponderarse si realmente la mejora que obtenemos en el código compensa sustancialmente la complejidad computacional que requiere. En general, suele dejarse esta decisión al programador mediante alguna directiva de compilación, ya que en la fase de desarrollo y depuración del programa la optimización puede ser un innecesaria, mientras que para la compilación del programa final puede resultar conveniente.

Estas mejoras se llevan a cabo mediante transformaciones en el código generado automaticamente por las fases anteriores del compilador de manera que mantenga su semántica original. En el caso general, la optimización de código puede dividirse en dos etapas, una optimización del código intermedio y una optimización del código máquina. vease fig.1.

Optimización de código intermedio

La inclusión de una fase de generación de código intermedio tiene la ventaja de facilitar la tarea de optimización, esto se debe a varios factores:

Cierto tipo de optimizaciones solo pueden llevarse a cabo si disponemos de la información necesaria en forma adecuada, como veremos mas adelante.

La optimización del código intermedio es independiente de la máquina destino y por tanto, es mas conveniente desde el punto de vista de la portabilidad del compilador.

Las técnicas actuales de optimización de código intermedio se basan en el análisis del flujo de control del programa y del flujo de datos.Una vez realizado este análisis se efectuan las oprtunas transformaciones.

El flujo de control del programa se obtiene mediante la identificación de los llamados "bloques básicos". Desde el punto de vista del flujo de control, las optimizaciones pueden ser locales o globales, dependiendo de que se realicen dentro de un mismo bloque básico, o que afecten a varios.