

1.3 Comprobaciones de tipos en expresiones

La labor de comprobación de tipos consiste en conferir a las construcciones sintácticas del lenguaje la semántica de tipificación y en realizar todo tipo de comprobaciones de dicha índole. Por su naturaleza, sin embargo, ésta se encuentra repartida entre la fase de análisis semántico y la generación de código intermedio.

Comprobaciones estáticas

Las comprobaciones estáticas recogen el compendio de todas aquellas tareas de carácter semántico que, por su naturaleza, pueden ser realizadas directamente durante la fase de compilación mediante el uso de los artefactos y mecanismos propios de dicha fase. Este tipo de comprobaciones son beneficiosas puesto que confieren seguridad a la ejecución del programa.

Características

- Diferente de la dinámica en runtime.
- Ejemplo: comprobación de tipos, flujo de control, unicidad.

Comprobaciones dinámicas

Las comprobaciones dinámicas son aquellas que no se realizan durante la fase de compilación y se delegan al momento de la ejecución del programa. Ello requiere generar código ejecutable específicamente diseñado para realizar tales comprobaciones. Los lenguajes con una carga excesiva de comprobaciones dinámicas generan programas más largos, lentos e inseguros en ejecución.

Verificación de tipos

Comprueba la compatibilidad de tipos de todas las expresiones del código fuente recuperando la información durante la gestión de declaraciones. Además se asegura de que no existe en el programa ninguna referencia a ningún símbolo no declarado.

Inferencia de tipos

En lenguajes sin tipificación de variables o con sobrecarga se aplican tareas de inferencia de tipos en el nivel gramatical de las expresiones para resolver el tipo de datos de la expresión resultante en función del contexto de evaluación.

Verificación de tipos

- Se puede hacer en forma
 - Estática: en tiempo de compilación
 - Dinámica: en tiempo de ejecución
 - Verifica:
 - Compatibilidad entre un operador y sus operandos
 - Flujo de control sea correcto
 - Duplicidad de nombres cuando esto sea prohibido
-

Verificación de tipos

- Puede hacerse durante el análisis sintáctico
 - Puede requerir una pasada sobre el árbol de análisis sintáctico entre el análisis sintáctico y la generación de código
-

Verificación de tipos

- Sobrecarga de operadores: el operador cambia su significado dependiendo del contexto
 - Coacción de tipos: el lenguaje brinda un operador que modifica el tipo de su operando
 - Funciones polimórficas: realizan un mismo procesamiento sobre elementos de distintos tipos
-

Sistema de Tipos

- Es una colección de reglas para asignar expresiones de tipo a diferentes partes de un programa
 - Para construirlo se necesita conocer:
 - Las construcciones sintácticas del lenguaje
 - La noción de tipos
 - Normalmente hay tipos básicos y tipos contruidos
-

Expresiones de tipo

- Sirven para tipar construcciones del lenguaje que tienen tipo construido
 - Definición:
 - Un tipo básico es una expresión de tipo
 - Un nombre de tipo es una expresión de tipo
 - Un constructor de tipo aplicado a una expresión de tipo, es una expresión de tipo
 - Una expresión de tipo se puede representar mediante un grafo
-