Tarea 2

Compiladores

**La Tabla de Símbolos.**

La tabla de símbolos puede iniciarse con cierta información útil, tal como:

- Constantes: PI, E, etc.

- Funciones de librería: EXP, LOG, etc.

- Palabras reservadas. Esto facilita el trabajo al lexicográfico, que tras reconocer un identificador lo busca en la tabla de símbolos, y si es palabra reservada devuelve un token asociado. Bien estructurado puede ser una alternativa más eficiente al tal y como lo hemos visto (hash perfecto).

Conforme van apareciendo nuevas declaraciones de identificadores, el analizador léxico, o el analizador sintáctico según la estrategia que sigamos, insertará nuevas entradas en la tabla de símbolos, evitando siempre la existencia de entradas repetidas.

El analizador semántico efectúa las comprobaciones sensibles al contexto gracias a la tabla de símbolos, y el generador de código intermedio usa las direcciones de memoria asociadas a cada identificador en la tabla de símbolos, al igual que el generador de código.

El optimizador de código no necesita hacer uso de ella

**Estructura de un compilador y relación a la Tabla de Símbolos**

El analizador semántico efectúa las comprobaciones sensibles al contexto gracias a la tabla de símbolos, y el generador de código intermedio usa las direcciones de memoria asociadas a cada identificador en la tabla de símbolos, al igual que el generador de código.

La tabla de símbolos contiene información útil para poder compilar, por tanto, existe entiempo de compilación, y no de ejecución. Sin embargo, en un intérprete, dado que la compilación y ejecución se producen a la vez, la tabla de símbolos permanece todo el tiempo.

**Gramáticas y lenguajes**

Dado un alfabeto Σ, un lenguaje es un conjunto (finito o infinito) de cadenas de símbolos pertenecientes analfabeto. Es posible que la cadena vacía, pertenezca a un lenguaje unas gramáticas una forma compacta de representar un lenguaje

IMPORTANTE:

Una gramática genera un único lenguaje, pero un mismo lenguaje puede ser generado por varias gramáticas

**Tipos de gramáticas se usan en los compiladores**

En los compiladores se utilizan solamente gramáticas regulares y gramáticas independientes del contexto (GIC)

Las gramáticas regulares se utilizan para especificar los tokens (en realidad, se utilizan expresiones regulares,

pero son equivalentes)

Las GIC se utilizan para especificar la sintaxis de las construcciones del lenguaje fuente

En los lenguajes de programación hay restricciones semánticas (p.ej. es necesario haber declarado una variable

Antes de utilizarla), que hacen que en realidad los lenguajes de programación sean lenguajes sensibles al contexto,

pero no se utilizan gramáticas sensibles al contexto, se utilizan GIC a las que se a ̃naden acciones para la

Comprobación de las restricciones semánticas