

Estructuras de datos estáticas

Autor: Moreno Madrona, Natividad (Ingeniera Técnica en Informática de Gestión e Ingeniera Técnica en Informática de Sistemas, Profesor de Enseñanza Secundaria).

Público: Ingeniería en Informática. **Materia:** Programación en lenguajes estructurados. **Idioma:** Español.

Título: Estructuras de datos estáticas.

Resumen

Las computadoras son máquinas dedicadas al manejo de información. Esta información se estructura de forma adecuada para obtener un rendimiento razonable en su memorización, tratamiento y recuperación. En este tema comenzamos estudiando el concepto de dato lógico, describiendo algunos de los tipos de datos más usuales. A continuación, estudiaremos algunas estructuras de datos utilizadas en programación, en sistemas operativos, o en el diseño físico de computadoras. Cuando utilizamos una computadora para resolver un problema debemos hacer una abstracción de la información y de las magnitudes que influyen en ella.

Palabras clave: Datos, programación, lenguajes.

Title: Static data structures.

Abstract

Computers are machines dedicated to handling information. This information is structured properly to get a reasonable return on their storage, treatment and recovery. This topic began studying the concept of logical data, describing some of the most common types of data. Then we study some data structures used in programming, operating systems, or the physical design of computers. When we use a computer to solve a problem we must make an abstraction of information and variables that influence it.

Keywords: Data, programming, languages.

Recibido 2016-04-19; Aceptado 2016-04-21; Publicado 2016-05-25; Código PD: 071047

1. INTRODUCCIÓN

Las computadoras son máquinas dedicadas al manejo de información. Esta información se estructura de forma adecuada para obtener un rendimiento razonable en su memorización, tratamiento y recuperación.

En este tema comenzamos estudiando el concepto de dato lógico, describiendo algunos de los tipos de datos más usuales.

A continuación, estudiaremos algunas estructuras de datos utilizadas en programación, en sistemas operativos, o en el diseño físico de computadoras.

2. TIPOS DE DATOS

Se denomina dato a cualquier objeto manipulable por la computadora. Cuando utilizamos una computadora para resolver un problema debemos hacer una abstracción de la información y de las magnitudes que influyen en ella. La representación de una magnitud como un dato se puede entender como una aplicación que hace corresponder un dato a cada valor de la magnitud.

Se denomina tipo de dato al conjunto de la transformación y de las operaciones y funciones internas y externas definidas sobre el conjunto de datos.

2.1. Datos de tipo entero

El tipo entero es una representación del conjunto de los números enteros. La representación es posible para un subrango de magnitudes enteras centrado en el origen: números entre $2^{n-1}-1$ y -2^{n-1} . La razón de esta limitación está en la necesidad de utilizar un espacio finito y fijo para cada dato, y en el hecho de que la transformación realizada entre los números enteros y el tipo de datos consiste en representar el número en binario y almacenarlo con un número fijo de bits.

2.2. Datos de tipo real

El tipo de datos real es una representación del conjunto de los números reales. Esencialmente, la transformación realizada consisten expresar el número de la forma $N = m \cdot B^e$ donde N es el número real a representar, B es la base utilizada para el exponente, e es el exponente del número y m es la mantisa. El número se almacena en la computadora yuxtaponiendo el signo, el exponente y la mantisa.

Usualmente, en una computadora se pueden utilizar diversas representaciones para los datos reales: simple, double o cuádruple precisión. En cada uno de estos tipos el número de bits dedicados a representar la mantisa, y a veces, también el exponente es diferente.

2.3. Datos de tipo lógico

Los datos de tipo lógico representan valores lógicos o booleanos. Pueden tomar uno de entre dos valores: verdadero o falso (V o F).

2.4. Datos de tipo carácter

Los datos de tipo carácter representan elementos individuales de conjuntos finitos y ordenados de caracteres. El conjunto de caracteres representado depende de la computadora. Uno de los conjuntos más usuales es el ASCII.

2.5. Datos de tipo enumerado

Los datos de tipo enumerado se definen explícitamente dando un conjunto finito de valores. No es un tipo normalizado.

Los tipos de datos vistos en secciones anteriores son usualmente tratados por la computadora a nivel de hardware. Mientras que los tipos de datos enumerados y subrango sólo son interpretados por el software. Internamente, los datos de tipo enumerado se almacenan como valores enteros. A cada valor se le asocia un entero consecutivo, comenzando por cero.

2.6. Datos de tipo subrango

El tipo subrango se define a partir del entero, carácter o de un tipo enumerado. Un dato de tipo subrango puede tomar determinados valores del tipo original (a partir de cual se ha definido el subrango) entre un mínimo y un máximo.

3. ESTRUCTURAS DE DATOS

Los tipos de datos anteriores se suelen denominar elementales, se pueden utilizar para construir otros más elaborados. Una estructura de datos o tipos de datos estructurado es un tipo de datos construido a partir de otros tipos de datos.

Una estructura de datos se dice que es homogénea cuando todos los datos elementales que la forman son del mismo tipo. En caso contrario se dice que es heterogénea.

Algunos tipos estructurados (listas y árboles) se declaran sin especificar el número de componentes que van a tener. En este caso el compilador les reserva el espacio de memoria mínimo que necesitan. Durante la ejecución del programa la estructura de datos puede ir creciendo. Una estructura de datos que es gestionada de esta forma se dice que es dinámica, por el contrario, una estructura de datos que siempre ocupa el mismo espacio se dice que es estática.

3.1. Arrays // (estática, homogénea)

Un array es una estructura de datos formada por una cantidad fija de datos de un mismo tipo, cada uno de los cuales tiene asociado uno o más índices que determinan de forma unívoca la posición del array. Cada índice es un dato de tipo subrango.

En general, al número de índices del array se le denomina número de dimensiones del array. Un array es una estructura de datos estática. Esto es, al definirla se especifica el número de elementos que la construyen. Este dato lo utiliza el

compilador para reservar el espacio necesario para almacenarla. Las matrices se almacenan en memoria ocupando un área contigua. Cada elemento ocupa el mismo número de palabras, que será el que corresponde al tipo de éstos.

3.2. Cadenas de caracteres

Una cadena de caracteres (String) es una estructura de datos formada por una secuencia de caracteres en un orden determinado. Las constantes de este tipo suelen ir entre comillas.

Sobre datos de tipo cadena de caracteres se pueden realizar las siguientes operaciones:

- Concatenar dos cadenas.
- Extracción de una subcadena.
- Comparación de dos cadenas.
- Obtención de la longitud de una cadena.

Aunque la cadena de caracteres y un array lineal de caracteres pueden contener la misma información, son tipos de datos distintos. Representan objetos distintos y permiten realizar operaciones distintas.

3.3. Registros o estructuras // (estática, heterogénea)

Un registro es una estructura de datos formada por yuxtaposición de elementos que contienen información relativa a un mismo ente. A los elementos que componen el registro se les denomina campos. Cada campo es de un determinado tipo. Para definir un registro es necesario especificar el nombre y el tipo de cada campo. Los campos pueden ser de un tipo estructurado.

Con los registros se pueden realizar asignaciones del registro completo a una variable de tipo registro, definida con los mismos campos en el mismo orden. Se puede realizar, al igual que en arrays, la selección de un campo. Esto se realiza especificando el nombre del campo.

Los registros se mapean en la memoria de una computadora por una simple yuxtaposición de sus componentes. La dirección de un componente (campo) i relativa a la dirección de origen del registro r se llama composición K_i del campo. Ésta se calcula como $K_i = S_1 + S_2 + \dots + S_{i-1}$, donde S_j es el tamaño (en palabras) del j -ésimo componente.

En lenguaje C, una estructura se define como:

```
Struct T
{
    T1 S1;
    T2 S2;
    ...
    Tn Sn;
}
```

3.4. Secuencias o ficheros

Todos los elementos de una secuencia son del mismo tipo, llamado tipo base T_0 de la secuencia.

Una secuencia s la representaremos con n elementos por medio de

$$S = \langle s_0, s_1, s_2, \dots, s_{n-1} \rangle$$

Donde n se le denomina longitud de la secuencia. Esta estructura parece ser muy semejante al array. La diferencia esencial es que, en el caso del array, el número de elementos es fijado por la declaración del array, en cambio, en la secuencia se deja abierto. Esto implica que puede variar durante la ejecución del programa.

Una secuencia se inspecciona mediante acceso secuencial, es decir, para llegar a un elemento n se deben leer los $n-1$ elementos anteriores a éste. El acceso secuencial puede reforzarse ofreciendo un conjunto de operadores de secuencia que facilitan el acceso a las variables de las secuencias de forma directa. Algunos de estos operadores primitivos serían:

- open(s): define a s como una secuencia vacía, es decir, la secuencia de longitud 0.
- write(s,x): agrega un elemento con valor x al extremo de la secuencia s .
- reset(s): coloca la posición actual (indicador de posición) al inicio de s .
- read(s,x): almacena en la variable x el elemento que se encuentra en la posición actual, y adelanta la posición hasta el siguiente elemento.

Con mucha frecuencia, los sistemas operativos permiten accesos no secuenciales. En particular, permiten el posicionamiento de la secuencia en un sitio arbitrario en vez de sólo en el inicio.

Bibliografía

- Weiss, M.A.: Data Structures and Algorithm Analysis in C++, 4th Edition, Pearson/Addison Wesley, 2013.
- Hernández, Z.J. y otros: Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++, Thomson, 2005.
- Shaffer, Clifford A.: Data Structures and Algorithm Analysis in C++, Third Edition, Dover Publications, 2011.