

Tipos de Datos

Un tipo de dato es un conjunto de valores y una serie de operaciones que pueden ser aplicadas a estos valores. Básicamente, los tipos de datos se pueden clasificar en elementales y estructurados. Los elementales o no estructurados son aquellos que no pueden dividirse en componentes (i.e. atómicos en composición de tipos). Los estructurados están formados por componentes y pueden descomponerse en términos de los valores que lo constituyen. Es posible expresar los valores de 3 formas: constantes, nombres o identificadores y expresiones. Es importante destacar que un tipo determina una clase de valores que puede asumir una variable o expresión. Así, para cada tipo de dato es importante conocer la cantidad de memoria que ocupa (Complejidad en Espacio) y los valores que pueden tomar (Cardinalidad).

Tipos de Datos Elementales : Integer, Char, Real, Boolean, Range, Enum, Pointer

En el computador, los valores se codifican en memoria como secuencias de 0's y 1's. Asumiremos que la cantidad de memoria de un tipo de dato elementales T es $CM(T) = 1$. Y la cantidad de valores que ocupa se define como $Card(T)$. Para cada tipo de dato se puede definir lo siguiente:

- Valores que puede tomar
- Operaciones que puede realizar y Operadores que pueden emplear
- Representación en el computador
- Cardinalidad

Tomando como ejemplo, el tipo Integer (y haciendo el mismo análisis para los otros tipos de dato elementales):

- Valores: Subconjunto de \mathbb{N}
- Operaciones: adición, substracción, multiplicación, división entera, residuo, lógicos, relacionales y de asignación. Operadores: $>$, $<$, \geq , \leq , $==$, \neq , $bNot$, $bXor$, $bAnd$, bOr , not , and , or , $lShift$, $rShift$
- Representación: Si se emplean n bits comprende 2^n valores. Sin signo de 0 a 2^n ; con signo en complemento A2 de -2^{n-1} a 2^{n-1}
- Cardinalidad: $2^n - 1$ (porque hay doble representación del 0)

Tipos de Datos Estructurados : String, Arreglo, Registro

Para el caso del Arreglo unidimensional:

- Valores: Valores estructurados $(v_{}, v_{<li+1>}, \dots, v_{<ls>})$ y $v_{<k>}$ es del tipo base del arreglo
- Operaciones: Selectora $[]$ y Constructora
- Representación: Desde una dirección base en memoria, los valores consecutivos $v_{<k>}$ en memoria. Se tiene un total de $ls - li + 1$ elementos
- Fórmula de Acceso: Dada la dirección base $dirbase$ y el arreglo A del tipo $Tbase$, un elemento $A[k]$ se accede como $dirbase + (k - li) * CM(Tbase)$
- Cardinalidad: $\prod_{i=li}^{ls} Card(Tbase) = Card(Tbase)^{ls-li+1}$
- Cantidad de Memoria: $CM(Tbase) * (ls - li + 1)$

Para el caso del Arreglo bidimensional (mostrando solo lo diferente que el arreglo unidimensional):

- Representación: Se deriva de la misma forma que el tipo unidimensional
- Fórmula de Acceso: Dada la dirección base $dirbase$ y el arreglo A del tipo $Tbase$, un elemento $A[i, j]$ se accede como $dirbase + [(i - li_1) * (ls_2 - li_2 + 1) + (j - li_2)] * CM(Tbase)$
- Cardinalidad: $\prod_{i=li_1}^{ls_1} \prod_{j=li_2}^{ls_2} Card(Tbase) = Card(Tbase)^{(ls_1-li_1+1)*(ls_2-li_2+1)}$
- Cantidad de Memoria: $CM(Tbase) * (ls_1 - li_1 + 1) * (ls_2 - li_2 + 1)$

Para el caso de Registros:

- Valores: Valores estructurados (v_1, v_2, \dots, v_k) y v_i es del tipo elemental o estructurado T_i
- Operaciones: Selectora $.$ de la forma $< variable > . < campo >$
- Cardinalidad: $\prod_{i=li}^k Card(Campo_i)$
- Cantidad de Memoria: $\sum_{i=1}^k CM(Campo_i)$