

ESPECIFICACIÓN DE LOS TAD

- El objetivo de la especificación es describir el comportamiento del TAD; consta de dos partes:
 - la descripción matemática del conjunto de datos
 - la de las operaciones definidas en ciertos elementos de ese conjunto de datos.
- La especificación del TAD puede tener un enfoque *informal*, que describe los datos y las operaciones relacionadas en *lenguaje natural*.
- Otro enfoque mas riguroso, la *especificación formal*, supone suministrar un conjunto de *axiomas* que describen las operaciones en su aspecto *sintáctico* y *semántico*.

ESPECIFICACIÓN INFORMAL DE UN TAD

- Consta de dos partes:
 - Detallar en los datos del tipo los valores que pueden tomar.
 - Describir las operaciones relacionándolas con los datos.
- El formato que generalmente se emplea, primero especifica el nombre del TAD y los datos:
 - TAD *nombre del tipo* (valores y su descripción)
- A continuación cada una de las operaciones con sus argumentos, y una descripción funcional en lenguaje natural, con este formato:
 - Operación(argumentos)
 - Descripción funcional
- La especificación informal de un TAD tiene como objetivo describir los datos del tipo y las operaciones según la funcionalidad que tienen. No sigue normas rígidas al hacer la especificación, simplemente indica, de forma comprensible, la acción que realiza cada operación.

EJEMPLO: TIPO ABSTRACTO DE DATOS CONJUNTO:

- *TAD Conjunto* (colección de elementos sin duplicidades, pueden estar en cualquier orden, se usa para representar los conjuntos matemáticos con sus operaciones).
- *Operaciones*, se ponen las operaciones básicas sobre conjuntos:
 - *Conjuntovacio*. Crea un conjunto sin elementos.
 - *Añadir(Conjunto, elemento)*. Comprueba si el elemento forma parte del conjunto; en caso negativo, es añadido. La operación modifica al conjunto.
 - *Retirar(Conjunto, elemento)*. Si el elemento pertenece al conjunto, es eliminado de éste. La operación modifica al conjunto.
 - *Pertenece(Conjunto, elemento)*. Verifica si el elemento forma parte del conjunto, en cuyo caso devuelve *cierto*.
 - *Esvacio(Conjunto)*. Verifica si el conjunto no tiene elementos, en cuyo caso devuelve *cierto*.
 - *Cardinal(Conjunto)*. Devuelve el número de elementos del conjunto.
 - *Union(Conjunto, Conjunto)*. Realiza la operación matemática de la unión de dos conjuntos. La operación devuelve un conjunto con los elementos comunes y no comunes a los dos conjuntos.

ESPECIFICACIÓN FORMAL DE UN TAD(1)

- La especificación formal proporciona un conjunto de axiomas que describen el comportamiento de todas las operaciones. La descripción ha de incluir una parte de sintaxis, en cuanto a los tipos de los argumentos y al tipo del resultado, y una parte de semántica, donde se detalla la expresión del resultado que se obtiene para unos valores particulares de los argumentos. La especificación formal ha de ser lo bastante *potente* para que cumpla el objetivo de verificar la corrección de la implementación del TAD. El esquema que sigue consta de una cabecera con el nombre del TAD y los datos:
 - TAD nombre del tipo (valores que toma los datos del tipo)
- Le sigue la sintaxis de las operaciones, que lista las operaciones mostrando los tipos de los argumentos y el tipo del resultado:
 - **Sintaxis**
 - Operación(Tipo argumento, ...) -> Tipo resultado

ESPECIFICACIÓN FORMAL DE UN TAD(2)

- A continuación se explica la semántica de las operaciones. Ésta se construye dando unos valores particulares a los argumentos de las operaciones, a partir de los que se obtiene una expresión resultado. Éste puede tener referencias a tipos ya definidos, valores de tipo lógico o referencias a otras operaciones del propio TAD.
 - **Semántica**
 - Operación(valores particulares argumentos) \Rightarrow expresión resultado
- Al hacer una especificación formal, siempre hay operaciones definidas por sí mismas que se consideran *constructores* del TAD. Se puede decir que mediante estos constructores se generan todos los posibles valores del TAD. Normalmente, se elige como constructor la operación que inicializa (por ejemplo, *Conjuntovació* en el *TAD Conjunto*) y la operación que añade un dato o elemento (esta operación es común a la mayoría de los tipos abstractos de datos). Se acostumbra a marcar con un asterisco las operaciones que son constructores.

EJEMPLO: ESPECIFICACION FORMAL TAD CONJUNTO

- *TAD Conjunto*(colección de elementos sin duplicidades, pueden estar en cualquier orden, se usa para representar los conjuntos matemáticos con sus operaciones).
- Sintaxis:
 - *Conjuntovacio \rightarrow Conjunto
 - *Añadir(Conjunto, Elemento) \rightarrow Conjunto
 - Retirar(Conjunto, Elemento) \rightarrow Conjunto
 - Pertenece(Conjunto, Elemento) \rightarrow boolean
 - Esvacio(Conjunto) \rightarrow boolean
 - Cardinal(Conjunto) \rightarrow entero
 - Union(Conjunto, Conjunto) \rightarrow Conjunto

EJEMPLO: ESPECIFICACION FORMAL TAD CONJUNTO

Semántica $\forall e_1, e_2 \in \text{Elemento}$ y $\forall C, D \in \text{Conjunto}$

- $\text{Añadir}(\text{Añadir}(C, e_1), e_1) \Rightarrow \text{Añadir}(C, e_1)$
- $\text{Añadir}(\text{Añadir}(C, e_1), e_2) \Rightarrow \text{Añadir}(\text{Añadir}(C, e_2), e_1)$
- $\text{Retirar}(\text{Conjuntovacio}, e_1) \Rightarrow \text{Conjuntovacio}$
- $\text{Retirar}(\text{Añadir}(C, e_1), e_2) \Rightarrow$ **si** $e_1 = e_2$ **entonces** $\text{Retirar}(C, e_2)$ **sino** $\text{Añadir}(\text{Retirar}(C, e_2), e_1)$
- $\text{Pertenece}(\text{Conjuntovacio}, e_1) \Rightarrow \text{falso}$
- $\text{Pertenece}(\text{Añadir}(C, e_2), e_1) \Rightarrow$ **si** $e_1 = e_2$ **entonces** cierto **sino** $\text{Pertenece}(C, e_1)$
- $\text{Esvacio}(\text{Conjuntovacio}) \Rightarrow \text{cierto}$
- $\text{Esvacio}(\text{Añadir}(C, e_1)) \Rightarrow \text{falso}$
- $\text{Cardinal}(\text{Conjuntovacio}) \Rightarrow \text{Cero}$
- $\text{Cardinal}(\text{Añadir}(C, e_1)) \Rightarrow$ **si** $\text{Pertenece}(C, e_1)$ **entonces** $\text{Cardinal}(C)$ **sino** $1 + \text{Cardinal}(C)$
- $\text{Union}(\text{Conjuntovacio}, \text{Conjuntovacio}) \Rightarrow \text{Conjuntovacio}$
- $\text{Union}(\text{Conjuntovacio}, \text{Añadir}(C, e_1)) \Rightarrow \text{Añadir}(C, e_1)$
- $\text{Union}(\text{Añadir}(C, e_1), D) \Rightarrow \text{Añadir}(\text{Union}(C, D), e_1)$