

Documento de definición del TAD *tablaF*

Definición del TIPO

Nombre: tablaF

Descripción: Almacena números enteros junto con la frecuencia de aparición de cada uno de ellos

Características: Se utiliza para contar las veces en las que aparece cada número cuando se lee una secuencia de números enteros

Valores no admitidos: Caracteres que no representen números enteros válidos

Definición de las OPERACIONES

acción inicializar (**var** t: tablaF);

{

Parámetros:

t: tabla de frecuencias

Devuelve:

Precondiciones:

Efecto:

crea la tabla de frecuencias t vacía

Excepciones:

}

Definición de las OPERACIONES

acción añadir (**var** t : tablaF; n : entero);

{

Parámetros:

t : tabla de frecuencias a la que se le va a añadir un entero

Devuelve:

Precondiciones:

la tabla t ha de ser creada correctamente (con la operación *inicializa*). Supondremos que los parámetros que se pasan a una operación han sido correctamente creados mediante una operación generadora y, por tanto, se pueden omitir las precondiciones

Efecto:

añade a la tabla t el número n e inicializa a 1 su frecuencia de apariciones. Si n ya se encuentra en la tabla t , se incrementa en 1 su frecuencia. Además, esta operación irá ordenando la tabla en orden decreciente de frecuencias

Excepciones:

}

Definición de las OPERACIONES

función total (t: tablaF): entero;

{

Parámetros:

t: tabla de frecuencias

Devuelve:

el número de enteros distintos que hay en la tabla t

Precondiciones:

Efecto:

Excepciones:

}

Definición de las OPERACIONES

acción info (t: tablaF; i:entero; **var** n, frec: entero);

{

Parámetros:

t: tabla de frecuencias

i: posición i-ésima de la tabla t

n: entero que ocupa el i-ésimo lugar en la tabla t en orden decreciente de frecuencias

frec: número de veces que se encuentra n en la tabla t

Devuelve:

Precondiciones:

1 ≤ i ≤ total(t)

Efecto:

almacena en el parámetro n el número que se encuentra en la posición i de la tabla t, y almacena en frec la frecuencia con la que aparece dicho número

Excepciones:

}

Tipos Abstractos de Datos

- ❑ Siguiendo refinamiento: elección de la representación del tipo *tablaF* e implementación de las operaciones

tipo

elemento = **registro**

número, frec: entero;

fregistro;

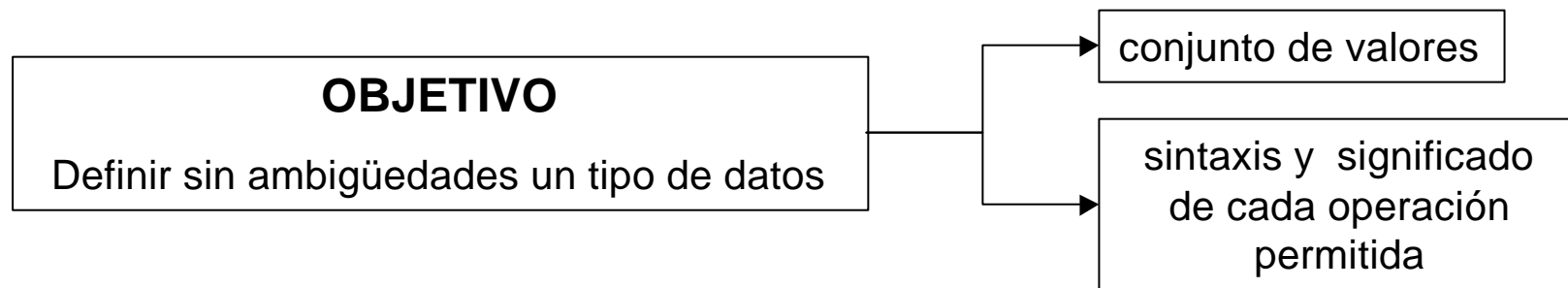
tablaF = **tabla** [1..max] **of** elemento;

- ❑ Distintas soluciones
- ❑ Ninguna debe influir en el código del algoritmo principal.
- ❑ Ejemplo: la ordenación se puede hacer en la operación *info* en lugar de ir ordenándola cada vez que se añade un elemento mediante la operación *añadir*, o sea, en la operación *info* se busca el de mayor frecuencia y se imprime

1.3 Especificación algebraica de un TAD

1.3.1 Introducción

- ❑ **Especificación algebraica.** Técnica formal para especificar tipos abstractos de datos



VENTAJAS

- ❑ Define tipos independientemente de cualquier representación e implementación
- ❑ Consigue unanimidad en la interpretación del tipo
- ❑ Posibilita la obtención de código automáticamente a partir de la especificación algebraica (aún no muy desarrollado)

1.3.2 Signatura de una especificación algebraica

❑ **Signatura.** Define los **géneros** y los nombres de las operaciones con sus **perfiles**

nombres de los nuevos tipos a especificar

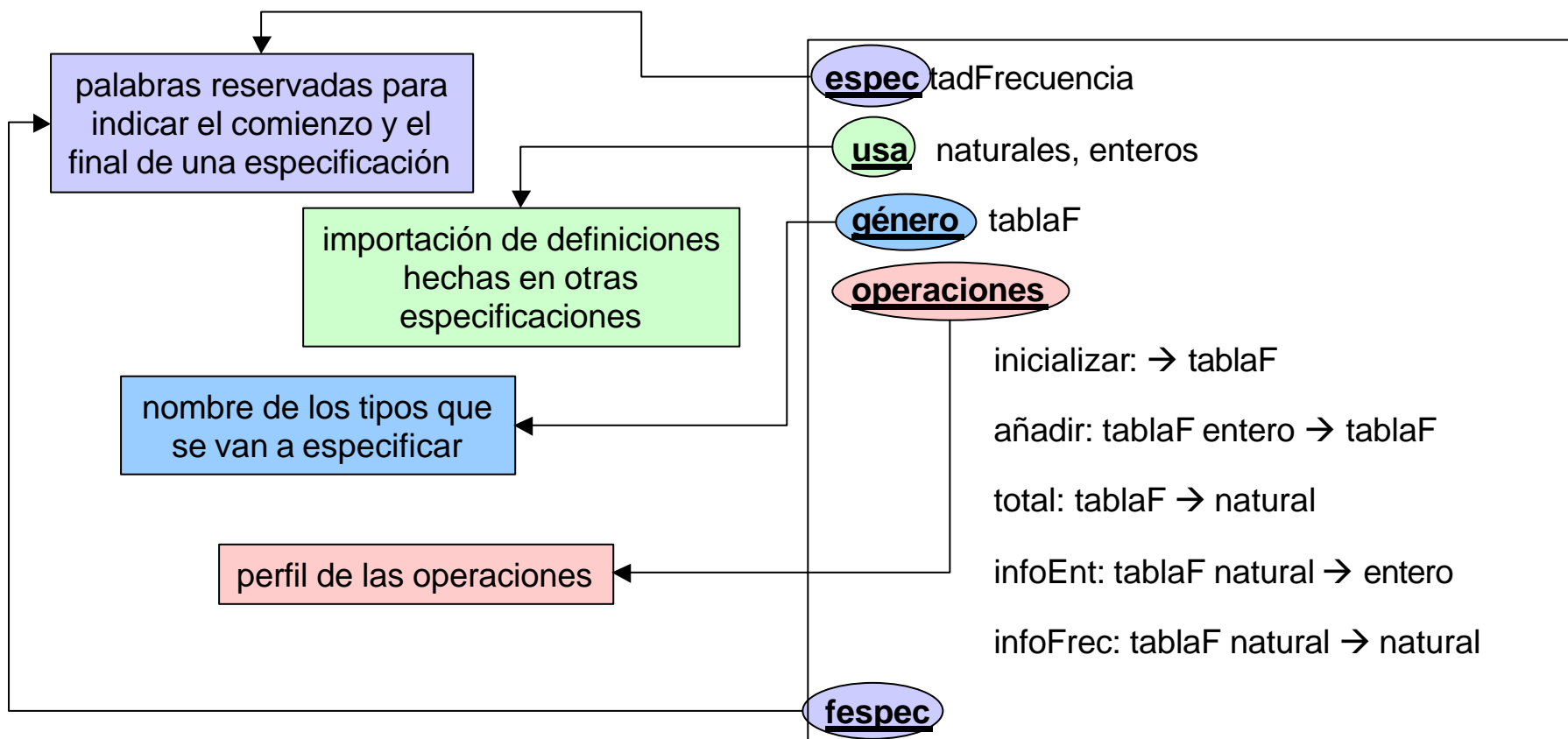
número y tipo de los parámetros, y tipo del resultado

CARACTERISTICAS

- ❑ Notación funcional
- ❑ Cada operación es una función con 0 o más parámetros
- ❑ Todas las operaciones devuelven un único valor de un tipo determinado

Tipos Abstractos de Datos

Ejemplo: Signatura del TAD *tablaF*



OTRAS CARACTERISTICAS

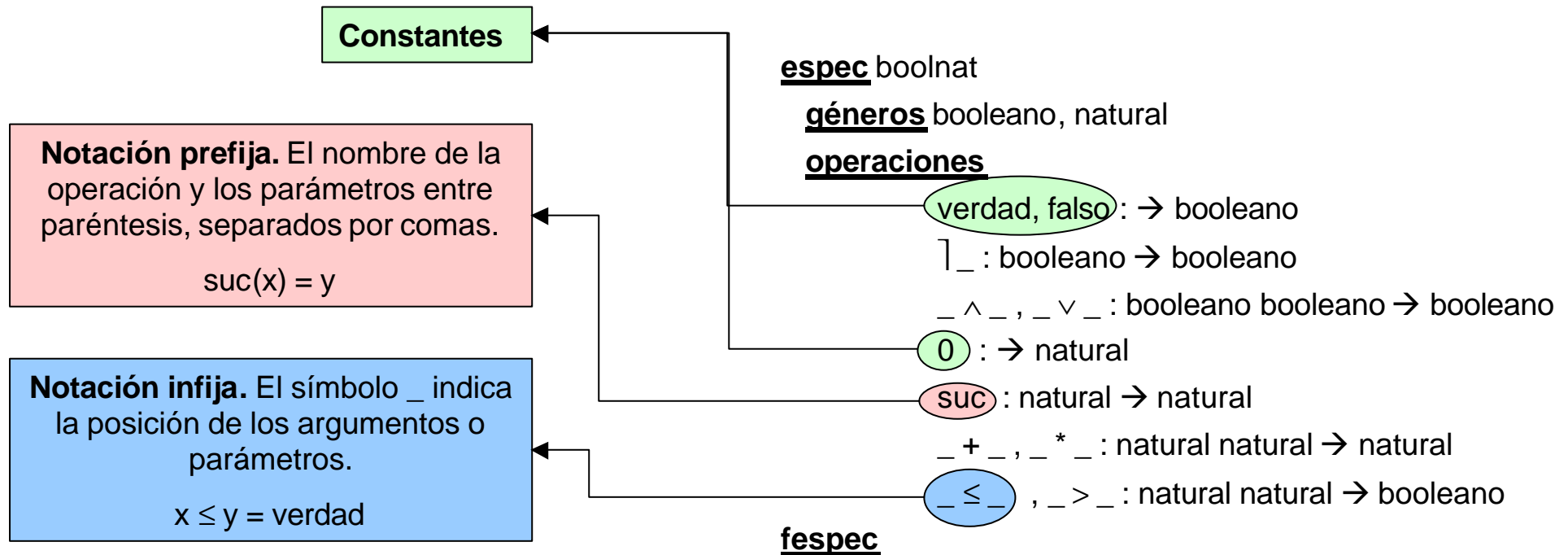
- ❑ Existen operaciones sin parámetros —————> **CONSTANTES**
 - `t:= inicializar();`
 - `inicializar(t);`

- ❑ La traducción de notación algebraica (funcional) a imperativa es inmediata
 - *total* se transforma en una función con el mismo perfil
 - *añadir* se transforma en un procedimiento

- ❑ La restricción a un solo resultado por función no es importante.
 - **acción** info (t: tablaF; i: entero; **var** n, frec: entero);

Tipos Abstractos de Datos

Ejemplo. Posible signatura de los tipos *booleanos* y *naturales* con el 0



1.3.3 Ecuaciones de una especificación algebraica

- ❑ También se conocen con el nombre de *axiomas*
- ❑ Determinan las propiedades y el comportamiento de las operaciones
- ❑ Toda especificación debe cumplir:
 - ⊙ sólo pertenecen al tipo los valores que puedan ser creados mediante términos sintácticamente correctos
 - ⊙ cada término bien formado denota un valor diferente del tipo especificado

Tipos Abstractos de Datos

Ejemplo. Queremos especificar el tipo *naturales con el 0*

espec naturales_1

género natural

operaciones

$0 : \rightarrow \text{natural}$

$\text{suc} : \text{natural} \rightarrow \text{natural}$

fespec

- los únicos valores que pueden construirse son:
 $0, \text{suc}(0), \text{suc}(\text{suc}(0)), \text{suc}(\text{suc}(\text{suc}(0))), \text{etc.}$
- cada término denota un valor diferente

Añadimos a la especificación la operación suma

espec naturales_2

género natural

operaciones

$0 : \rightarrow \text{natural}$

$\text{suc} : \text{natural} \rightarrow \text{natural}$

$_ + _ : \text{natural} \text{ natural} \rightarrow \text{natural}$

fespec

- Incumple la segunda propiedad.
- Los términos " $\text{suc}(0)$ " y " $0 + \text{suc}(0)$ " generan el natural 1

- ❑ 2 puntos de vista respecto al papel de las ecuaciones en las especificaciones algebraicas:
 - igualar términos que generan el mismo valor (punto de vista algebraico)
 - definir el comportamiento de las operaciones con todas las posibles combinaciones de valores (patrones) que pueden tomar sus parámetros (punto de vista semántico)

- ❑ Formato de una ecuación: **término_1 = término_2**
 - donde *término_1* y *término_2* son términos bien formados de un mismo género