Algoritmos
Estructuras de Datos I

Facultad de Ciencias Exactas y Tecnología

Universidad Nacional de Tucumán

2025

Especificación Algebraica(2)

ESPECIFICACION ALGEBRAICA
Ejemplo

TIPO: NAT

OPERACIONES

Sintaxis:

CERO : → NAT
SUCC : NAT→ NAT
IGUALCERO : NAT → BOOL
PRED : NAT - {CERO} → NAT   parcial

ESPAR : NAT → BOOL
IGUAL : NAT x NAT → BOOL

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: NAT</u>

OPERACIÓN MAX

Sintaxis:

MAX **:** NAT x NAT → NAT

Semántica: Para todo x , y $\in$ NAT

MAX (CERO,CERO) $\equiv$ CERO
MAX (CERO,SUCC(x)) $\equiv$ SUCC(x)
MAX (SUCC(x),CERO) $\equiv$ SUCC(x)
MAX (SUCC(x),SUCC(y)) $\equiv$ SUCC (MAX(x,y))

ESPECIFICACION ALGEBRAICA
Ejemplo

TIPO: NAT **Semántica:** Para todo x , y $\in$ NAT

IGUALCERO(CERO) $\equiv$ TRUE
IGUALCERO(SUCC(x)) $\equiv$ FALSE
PRED(SUCC(x)) $\equiv$ x
ESPAR(CERO) $\equiv$ TRUE
ESPAR (SUCC(CERO)) $\equiv$ FALSE
ESPAR (SUCC (SUCC(x))) $\equiv$ ESPAR(x)
IGUAL(CERO,CERO) $\equiv$ TRUE
IGUAL (CERO,SUCC(x)) $\equiv$ FALSE
IGUAL (SUCC(x),CERO) $\equiv$ FALSE
IGUAL (SUCC(x),SUCC(y)) $\equiv$ IGUAL (x,y)
MAX (CERO,CERO) $\equiv$ CERO
MAX (CERO,SUCC(x)) $\equiv$ SUCC(x)
MAX (SUCC(x),CERO) $\equiv$ SUCC(x)
MAX (SUCC(x),SUCC(y)) $\equiv$ SUCC (MAX(x,y))

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: NAT</u>

OPERACIÓN SUMA

Sintaxis:

SUMA **:** NAT x NAT → NAT

Semántica: Para todo x , y $\in$ NAT

SUMA (CERO,CERO) $\equiv$ CERO
SUMA (CERO,SUCC(x)) $\equiv$ SUCC(x)
SUMA (SUCC(x),CERO) $\equiv$ SUCC(x)
SUMA (SUCC(x),SUCC(y)) $\equiv$ SUCC(SUCC(SUMA(x,y)))

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: NAT</u>

OPERACIÓN SUMA en 2 axiomas

Sintaxis:

SUMA **:** NAT x NAT → NAT

Semántica: Para todo x , y $\in$ NAT

SUMA (CERO, y) $\equiv$ y

SUMA (SUCC(x), y) $\equiv$ SUCC(SUMA(x,y))

ESPECIFICACION ALGEBRAICA
Ejemplo

TIPO: NAT

OPERACIÓN MULT

Sintaxis:

MULT **:** NAT x NAT → NAT

Semántica: Para todo x , y $\in$ NAT

MULT (CERO,CERO) $\equiv$ CERO
MULT (CERO,SUCC(x)) $\equiv$ CERO
MULT (SUCC(x),CERO) $\equiv$ CERO
MULT (SUCC(x),SUCC(y)) $\equiv$
SUCC(SUMA(SUMA(MULT(x,y),x),y))

ESPECIFICACION ALGEBRAICA Ejemplo

TIPO: NAT

OPERACIÓN MULT en 2 axiomas

Sintaxis:

MULT **:** NAT x NAT → NAT

Semántica: Para todo x , y $\in$ NAT

MULT (x, CERO) $\equiv$ CERO

MULT (x, SUCC(y)) $\equiv$ SUMA(MULT(x,y),x)

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: NAT</u>

OPERACIÓN RESTA

Sintaxis:

RESTA **:** NAT x NAT → NAT U {indefinido}

Semántica: Para todo x , y $\in$ NAT

RESTA (CERO,CERO) $\equiv$ CERO
RESTA (CERO,SUCC(x)) $\equiv$ indefinido
RESTA (SUCC(x),CERO) $\equiv$ SUCC(x)
RESTA (SUCC(x),SUCC(y)) $\equiv$ RESTA(x,y)

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: CADENA</u>

OPERACIONES

Sintaxis:

NULA **:** → CADENA

AGREGAR **:** CADENA X CHAR → CADENA

ESNULA **:** CADENA → BOOL

LARGO **:** CADENA → ENTERO ≥ 0

CONCAT **:** CADENA X CADENA → CADENA

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: CADENA</u>

Semántica: Para todo s,t $\in$ CADENA, $\forall$ c $\in$ CHAR,

ESNULA(NULA) $\equiv$ TRUE

ESNULA(AGREGAR(s,c)) $\equiv$ FALSE

LARGO(NULA) $\equiv$ 0

LARGO(AGREGAR(s,c)) $\equiv$ LARGO(s) + 1

CONCAT(s,NULA) $\equiv$ s

CONCAT(s, AGREGAR(t,c)) $\equiv$ AGREGAR(CONCAT(s,t),c)

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: COMPLEJO</u>

OPERACIONES

Sintaxis:

ARMAR **:** REAL x REAL→ COMPLEJO

SUMA **:** COMPLEJO x COMPLEJO → COMPLEJO

RESTA **:** COMPLEJO x COMPLEJO → COMPLEJO

MULTIPLICA **:** COMPLEJO x COMPLEJO → COMPLEJO

DIVIDE **:** COMPLEJO x COMPLEJO → COMPLEJO U {indefinido}

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: COMPLEJO</u>

Sintaxis:

INVERSO : COMPLEJO → COMPLEJO U {indefinido}

OPUESTO : COMPLEJO → COMPLEJO

PREAL : COMPLEJO → REAL

PIMAG : COMPLEJO → REAL

ESREAL: COMPLEJO → BOOL

ESIMAG : COMPLEJO → BOOL

CONJUGADO : COMPLEJO → COMPLEJO

IGUAL : COMPLEJO x COMPLEJO → BOOL

NORMA : COMPLEJO → REAL

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: COMPLEJO</u>

Semántica: Para todo a, b, c, d $\in$ REAL,

SUMA(ARMAR(a,b), ARMAR(c,d)) $\equiv$ ARMAR (a+c,b+d)

RESTA(ARMAR(a,b), ARMAR(c,d)) $\equiv$ ARMAR (a-c,b-d)

MULTIPLICA(ARMAR(a,b), ARMAR(c,d)) $\equiv$ ARMAR (a*c-b*d, a*d+b*c)

DIVIDE(ARMAR(a,b), ARMAR(c,d)) $\equiv$

si c*c+d*d = 0 entonces

indefinido

sino

ARMAR ((a*c+b*d)/(c*c+d*d), (-a*d+b*c)/(c*c+d*d))

ESPECIFICACION ALGEBRAICA
Ejemplo

TIPO: COMPLEJO

Semántica: Para todo a, b, c, d $\in$ REAL,

INVERSO (ARMAR(a,b)) $\equiv$ si a**=**0 AND b**=**0 entonces indefinido
sino ARMAR (a**/**(a*****a**+**b*****b),**-**b**/**(a*****a**+**b*****b))

OPUESTO (ARMAR(a,b)) $\equiv$ ARMAR (**-**a,**-**b)

PREAL (ARMAR(a,b)) $\equiv$ a

PIMAG (ARMAR(a,b)) $\equiv$ b

ESREAL (ARMAR(a,b)) $\equiv$ b**=**0

ESIMAG (ARMAR(a,b)) $\equiv$ si a**=**0 AND b**≠**0 entonces TRUE sino FALSE

CONJUGADO(ARMAR(a,b)) $\equiv$ ARMAR (a,**-**b)

IGUAL(ARMAR(a,b), ARMAR(c,d)) $\equiv$ si a**=**c AND b**=**d entonces TRUE
sino FALSE

NORMA(ARMAR(a,b)) $\equiv$ a*****a **+** b*****b

Se usan los operadores de números reales,: **+** , **-** , ***** , **/** , **=** , **≠**

TIPOS ABSTRACTOS DE DATOS GENERICOS

- Los TADs genéricos representan colecciones de elementos todos del mismo tipo.
- Estos TADs definen un cierto comportamiento independiente del tipo de sus elementos.
- Para poder expresar genéricamente el tipo de los elementos se utilizan parámetros.
- De esta forma, se pueden construir ejemplares del TAD genérico utilizando otros TADs que cumplan con las restricciones del parámetro indicado en su especificación.

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: VECTOR (ITEM)</u>

OPERACIONES

Sintaxis:

VECTORVACIO **:** → VECTOR

ALMACENAR **:** VECTOR **x** ENTERO **x** ITEM → VECTOR

OBTENER **:** VECTOR **x** ENTERO → ITEM U { indefinido}

Semántica: Para todo A ∈ VECTOR, ∀ i,j ∈ ENTERO, ∀x ∈ ITEM

OBTENER(VECTORVACIO,i) ≡ indefinido

OBTENER(ALMACENAR(A,i,x), j) ≡ si i**=**j entonces

x

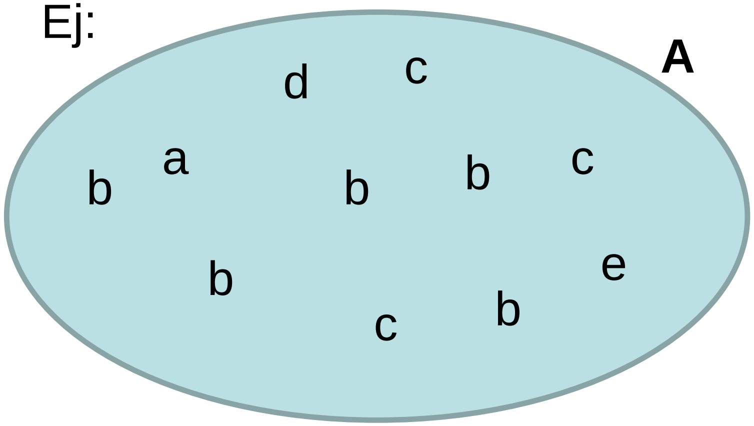
sino

OBTENER(A,j)

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: MULTICONJUNTO(ITEM)</u>

Ej:



OPERACIONES

Sintaxis:

MULTICONJUNTOVACIO **:** → MULTICONJUNTO

INSERTAR **:** MULTICONJUNTO **x** ITEM → MULTICONJUNTO

ESVACIO **:** MULTICONJUNTO → BOOL

PERTENECE **:** MULTICONJUNTO **x** ITEM → BOOL

BORRAR **:** MULTICONJUNTO **x** ITEM → MULTICONJUNTO

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: MULTICONJUNTO(ITEM)</u>

Semántica: Para todo A $\in$ MULTICONJUNTO , $\forall$ i, j $\in$ ITEM.

ESVACIO(MULTICONJUNTOVACIO) $\equiv$ TRUE
ESVACIO(INSERTAR(A,i)) $\equiv$ FALSE

PERTENECE(MULTICONJUNTOVACIO,i) $\equiv$ FALSE
PERTENECE(INSERTAR(A,i),j) $\equiv$ si i**=**j entonces
TRUE
Sino
PERTENECE(A,j)

Otra manera de definir PERTENECE:

PERTENECE(MULTICONJUNTOVACIO,i) $\equiv$ FALSE
PERTENECE(INSERTAR(A,i),j) $\equiv$ (i**=**j) OR PERTENECE(A,j)

= representa la operación IGUALITEM

ESPECIFICACION ALGEBRAICA
Ejemplo

<u>TIPO: MULTICONJUNTO(ITEM)</u>

Semántica: Para todo A $\in$ MULTICONJUNTO , $\forall$ i, j $\in$ ITEM.

BORRAR(MULTICONJUNTOVACIO,i) $\equiv$ MULTICONJUNTOVACIO

BORRAR(INSERTAR(A,i),j) $\equiv$ si i**=**j entonces

BORRAR (A,j)

sino

INSERTAR(BORRAR(A,j),i)

= *representa la operación IGUALITEM*

Observación: BORRAR borra todas las ocurrencias de un ITEM