

Tipos y subtipos

- Tipo de un objeto: dominio de valores que el objeto puede tomar y el conjunto de operaciones que se le puede aplicar
- Ejemplo: x es un entero
- dominio: $\dots, -2, -1, 0, 1, 2, 3, \dots$
- operaciones: $+, -, *, \dots$

Tipos y subtipos (2)

- Subtipo de un tipo: Un tipo $T1$ es subtipo de un tipo $T2$ si cualquier instancia de $T1$ es también instancia de $T2$
 - Ejemplo: Z es subtipo de R
- El concepto de subtipo implica que una instancia de $T1$ puede ser usada en el contexto de $T2$
 - Ejemplo: $A +_R B, A \text{ y } B \in Z$

Tipos y subtipos (3)

- Principio de sustitución
 - Refleja: $T1 \leq T1$
 - Transitiva: $T1 \leq T2$ y $T2 \leq T3 \implies T1 \leq T3$
 - Antisimétrica: $T1 \leq T2$ y $T2 \leq T1 \implies T1 = T2$
- Ejemplo: $N \leq Q \leq R$

Tipos y subtipos (4)

La relación de subtipos puede ser establecida a través de:

- subconjuntos
- subtipos de tipos estructurados (tuplas)
- subtipos de funciones

Subconjuntos como subtipos

- Conjuntos $\rightarrow$ Tipos
- Subconjuntos $\rightarrow$ Subtipos
 - $\text{dom}(T1) \subseteq \text{dom}(T2) \equiv T1 \leq T2$

Ejemplo: Sea Z el tipo de los enteros. Los siguientes son subconjuntos de Z :

- $S1 : 1..100$
- $S2 : \text{números primos}$
- $S3 : \text{números pares}$
- ¿Principio de substitución? $S1, S2, S3$
- ¿Cerrado? $S3$ con respecto a la $+, -, *$

Subconjuntos como subtipos (2)

- Subtipo completo: T1 es un subtipo completo de T2 si los operadores de T2 se comportan en forma compatible con argumentos de T1. Ejemplo: S3
- La relación de subtipos puede ser mantenida, además, con:
 - Restricciones para obligar cerradura
 - Heredando los operadores sin exigir cerradura

Subtipos de tipos estructurados

Tipos estructurados como tuplas

- Sean $T_1, T_2, \dots, T_n$ tipos $a_1, a_2, \dots, a_n$ y nombres de atributos.
- Se define $[a_1 : T_1, a_2 : T_2, \dots, a_n : T_n]$ tipo estructurado (tupla) como el conjunto de todas las tuplas que al menos tienen atributos $a_1, a_2, \dots, a_n$, cuyos valores están en $\text{dom}(T_1), \text{dom}(T_2), \dots, \text{dom}(T_n)$

■ Ejemplo:

- **Persona**: [nombre:string, edad:int:, direccion:string]
- **Persona** incluye:

[nombre:"Bruno", edad:5, direccion:"xxx"]

[nombre:"Catalina", edad:14, direccion:"yyy", fechanacimiento:"06-02-1994"]

Subtipos de tipos estructurados (2)

Relación de subtipo entre tuplas

Si $T_i \leq U_i \Rightarrow$

$[a_1 : T_1, a_2 : T_2, \dots, a_m : T_m] \leq [a_1 : U_1, a_2 : U_2, \dots, a_n : T_n], m \geq n$

● Ejemplo:

[nombre:string		[nombre:string
edad: Z		edad: Z
dirección:string	≤	direccion:string]
fechanacimiento:string]		

Colecciones: $S1\{t1\} \leq S2\{t2\} \Rightarrow t1 \leq t2$

■ Tipos de datos paramétricos

Ej: $\text{Lista } \langle int \rangle \leq \text{Lista } \langle float \rangle$

Contraste entre herencia y subtipos

¿Cuándo un *TDA1* es subtipo de otro *TDA2*?

- Estructura del *TDA1* (variables de instancia) subtipo de la estructura del *TDA2*
- Comportamiento del *TDA1* (firmas de sus métodos) debe ser consistente con el comportamiento de *TDA2*

¿Cuál es más importante?

- La relación de subtipos en el comportamiento

¿Cómo se representa el comportamiento?

- Firma: nombre del método, tipos de los argumentos y del valor retornado

Contraste entre herencia y subtipos(2)

Definición de subtipos:

TDA1 es subtipo de *TDA2* si cada método *M* del *TDA2* tiene un método en el *TDA1*

- Ejemplo: Conjunto es subtipo de Bolsa
 - Conjunto: no contiene elementos
 - Bolsa: contiene elementos repetidos
- Operaciones: intersección, unión, resta, pertenece, entre otras
- Implementación: cada uno puede tener su propia representación

Subtipos implícitos versus herencia explícita

Una relación de subtipos se puede definir en forma:

- Explícita (herencia)
- Implícita (deducción a partir de las propiedades)
- Esquemas híbridos (tipos de datos parametrizados)

Subtipos y enlace dinámico

- Enlace dinámico se produce en:
 - objetos (tipo desconocido en tiempo de compilación.
Esto ocurre en lenguajes sin declaración de variables)
 - métodos (operaciones)
- ¿Qué encontramos en los lenguajes?
 - Definición de tipos de variables en forma estática (C+, Eiffel, Java)
 - No hay declaración de tipos de variables (Smalltalk)