Principios de Lenguajes de Programación

Tipo de Datos:
Introducción
Tipos de datos elementales

Facultad de Informática
Universidad Nacional del Comahue

Primer Cuatrimestre

Índice

Unidad 2. Objeto, Valor y Tipos de Datos.

Concepto de Tipo de Datos. Mecanismos de definición de tipos. Tipos primitivos. Tipos compuestos y recursivos. Sistemas de Tipos. Equivalencia, Compatibilidad, Chequeo y Cohesión. Implementación de Objetos de Datos. Impacto en el diseño de los lenguajes de programación.

Índice

- Tipo de Datos
- Especificación e implementación
- Chequeo de tipo
- Lenguajes fuertemente tipados
- Conversiones de Tipos
- Tipos Elementales / Primitivos

Objeto de Dato

Un agrupación de una o más piezas de datos en una computadora virtual, que tiene entidad en ejecución

- Objetos de datos definidos por el programador:
- Objetos de datos definidos por el sistema:

Objeto de Dato

Un agrupación de una o más piezas de datos en una computadora virtual, que tiene entidad en ejecución

- Objetos de datos definidos por el programador: variables, constantes, arreglos, archivos etc.
- Objetos de datos definidos por el sistema: pilas de almacenamiento en tiempo de ejec. Memorias intermedia de archivo, lista de espacio libre

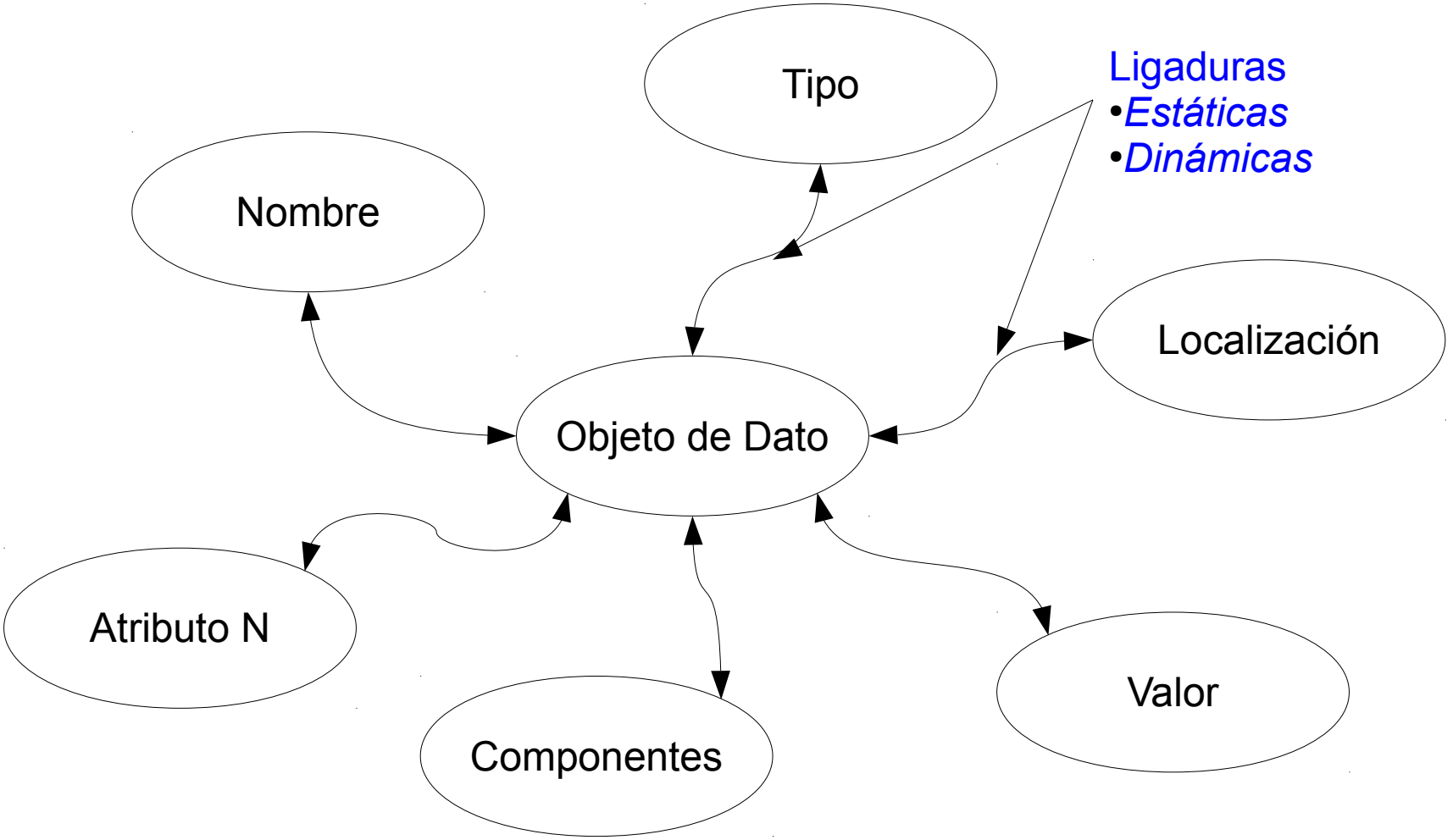
Objetos de Dato

- Los atributos de los OD
  - Permiten caracterizarlos
  - Descriptor: conjunto de atributos de un objeto de dato
  - Dependientes del lenguaje y la estructura asociada
    - Por ejemplo: Arreglos heterogéneos u homogéneos, fijos o de tamaño variante, con chequeo de rango, etc.
- OD Elementales
  - Contiene un valor que es manipulado como una unidad
- OD Compuestos / Estructurados
  - Combinan objetos de datos elementales

Objetos de Dato

- Se caracterizan por las ligaduras que pueden establecer con los atriburos
  - Ligaduras dinámicas
  - Ligaduras estáticas

Objeto de Dato



Objetos de Dato

- Variables: valor modificable (us. asignación)
- Constantes: OD enlazado en forma permanente a su valor o valores
  - Constante Literal
  - Definida por el programador
  - Tiempo de la ligadura

Tipo de Dato

Un **tipo de dato** define el conjunto de OD y el conjunto de operaciones definidas para manipular estos objetos

- Un OD representa una instancia de un tipo definido por el usuario (usualmente dato abstracto)

Tipo de Dato

Elementos básicos de una **especificación** de un TD

- **Atributos:** que distinguen OD de ese TD
- **Valores posibles:** que un OD de un TD tiene
- **Operaciones asociadas:** manipulaciones de OD de ese tipo

Ej: TD array

- Atributos:
- Valores:
- Operaciones:

Tipo de Dato

Elementos básicos de una **especificación** de un TD

- **Atributos:** que distinguen OD de ese TD
- **Valores posibles:** que un OD de un TD tiene
- **Operaciones asociadas:** manipulaciones de OD de ese tipo

Ej: TD array

- Atributos: #dimensión, rango de los subíndices p/u dimensiones, TD componentes
- Valores:conj de # que son valores válidos p/el array
- Operaciones: selección de un elem del array, crear array, acceder a su limite sup e inferior, etc.

Tipo de Dato

Elem. básicos de una **implementación** de un TD

- **Representación**/Almacenamiento del dato
- **Implementación** propia de las **operaciones** que manipulan la representación

Especificación del TD Elemental

- **OD Elemental:** contiene un valor de dato simple.
- Tipo de Dato Elemental:
    - Clase de OD Elem. En la que se definen op
- Lges tienden a tener un conj de TPElem: Integer, real, character, boolean, enumerado y puntero.
    - Sus especificaciones pueden ser distintas según los lenguajes

Tipo de Dato

Elem. básicos de una **implementación** de un TD

- **Representación**/Almacenamiento del dato
- **Implementación** propia de las **operaciones** que manipulan la representación

Especificación, implementación, representación sintáctica y verificación de tipo de datos.

Tipo de Dato: especificación TD Elemental

- Atributos
  - Tipo y Nombre (invariantes durante su tiempo de vida)
  - Algunos atributos se Incorporados al descriptor (en ejecución)
  - Otros son usados sólo para diseñar representación del almacenamiento, no explícitamente en ejecución
  - Valor de atributo <> Valor del OD (rep explicita en ejec)
- Valores
  - El tipo determina el conjunto de valores que puede ligarse a un objeto de dato. 1 solo valor
    - Ej, C int short long char (los caracteres se guardan como enteros de 8 bits en el tipo char).
  - Usualmente un conjunto ordenado (valor menor y mayor)

Tipo de Dato: especificación TD Elemental

- Operaciones
  - Posibles operaciones del objeto de un tipo de dato
    - Primitivas, parte del lenguaje
    - Definidas por programador como subprogramas, métodos de clase, etc.
  - Operaciones
    - Fc. Mat: argumentos y resultados
    - Dominio, conjunto de valores de entrada válidos
    - Rango, conjunto de resultados posibles
    - Acción, cómo se produce el resultado

Tipo de dato: especificación

- Especificación de la Signatura de la Operación
  - Especifica dominio y rango

    Número, orden y tipo de los argumentos del dominio
  - Número, orden y tipo del rango

```
NomOp: tipo1 x tipo2 x … x tipoN → tipo_Resultado
  + : entero x entero → entero
  = : entero x entero → boolean
  SQRT : real → real
```

  - Aridad de la operación

  - Acción:
    - Especificada por una semántica formal o
    - Especificada durante la implementación

Tipo de dato: especificación

- Problemas con las especif. formal de las op.
  - No definidas para algunas entrada
  - Argumentos Implícitos, ej, uso de variables globales
  - Efectos colaterales: Resultados implícitos, la operación modifican otros OD
  - Auto modificable (sensible a la historia), con cambios de datos locales entre llamadas
    - Ej. función random modifica la semilla (seed)

Tipo de Dato

- Implementación de un TD: representación + algoritmo/s
  - Representación del Almacenamiento del dato
    - Atributos:
      - Representación que el hw proporciona:
        - Eficientes en almacenamiento y velocidad. C, FORTRAN, PASCAL
      - Descriptor:
        - se pueden guardar en el descriptor como parte del OD. Flexibilidad. LISP, PROLOG

Tipo de Dato: implementación

- Implementación propia de las operaciones
  - *Operaciones de Hardware*
    - Implementación directa, ej suma entera
  - *Funciones/Subprogramas*
    - Cálculo de raíz cuadrada en otras representaciones
  - *Código in-line*
    - Simil subprograma, con código copiado
    - Levemente más eficiente

Tipo de Dato: algunos conceptos

- Chequeo/Control de Tipo
- Tipado Fuerte
- Conversiones de tipo

Chequeo o verificación de Tipo

- Chequeo/verificación de cada operación que ejecuta el programa, para verificar el número correcto de argumentos con los tipos de dato apropiados
- *Chequeo de Tipo Dinámico*
  - En ejecución.
    - Marcas de tipo
    - Ej. lisp Prolog
- *Chequeo de Tipo Estático*
  - Durante la compilación (antes de la ejecución)

Chequeo de Tipo Dinámico

- Ventaja:
  - Flexibilidad
- Desventajas:
  - Se dificulta la depuración (debugging), algunas alternativas de ejecución nunca se controlan
  - Almacenamiento extra para la información del tipo durante la ejecución del programa
  - Se simula el chequeo por software, lo que reduce la velocidad de ejecución

Chequeo de Tipo Estático

- Ventaja:
  - Mayor velocidad de ejecución
- Requiere:
  - Para cada operación, el número, orden y tipo de dato de cada operando y del resultado
  - El tipo de cada objeto de dato:
    - Variable, usualmente con nombre
      - Siempre con el mismo tipo
    - Constante definida y literal

Tipado fuerte

- Si se detecta todos los errores sintáctico estáticamente
  - Una función **f**, con signatura **f: S → R**, es seguro con respecto al tipo si cualquier ejecución de **f** no puede generar valores fuera de **R**
  - Si todas las op son seguras en cto a tipos, el lge es fuertemente tipado

Inferencia de Tipo

- Para tipos de dato implícitos, se usa si la interpretación no es ambigua
- Ej ML

```
fun area(long:int,ancho:int):int=long*ancho
```

Inferencia de Tipo

- `fun square(x)  :  real = x * x;`
  - La función devuelve valor real, luego el parámetro es real
  - X ahora es real
- Ejemplo,
  - ML no permite funciones sobrecargadas
  - No pueden existir square(x) para enteros y otra para real
- Otras formas de especificar un parámetro real:
  - `fun square(x : real) = x * x;`
  - `fun square(x) = (x : real) * x;`
  - `fun square(x) = x * (x : real);`

Conversión de Tipo

- Un error de tipo (type mismatch) genera:
  - Error
  - Conversión de Tipo
- Conversión de tipo:

```
Op-Conversión: Tipo1 → Tipo2
```

  - ***Implícita****:* o Coerción, realizada por el sistema
  - ***Explícita****:* rutina que cambia de un tipo de dato a otro
- *Coerciones:*
  - *Con o sin pérdida de información*
    - Promoción o Ensanchamiento
    - Degradación o Estrechamiento

Tipado fuerte-Conclusión

- Un lenguaje de programación es de tipado fuerte si siempre se pueden detectar los errores originados por tipos
  - Los tipos de todos los operandos se pueden conocer, ya sea en compilación o en ejecución
- Ventaja:
  - Permite la detección temprana de los usos incorrectos de variables que resultan de errores de tipo

Conversión de Tipo

- Dos filosofías opuestas:
  - No hay coerciones (Ada)
  - Coerción es la regla (lenguaje C)
- Ejemplo de Lenguajes
  - Pascal
    - Conversión Explícita, ej, con función *round*
    - Conversión Implícita, en operaciones de enteros y reales
  - C
    - Conversión Explícita, con casting (int) X para X float

Conversión de Tipo: coerción

- Ventajas:
  - Libera al programador de algunas consideraciones de bajo nivel por ejemplo, suma de enteros y reales
- Desventajas:
  - Puede “esconder” errores de programación serios

Tipo de Dato

- Tipos de Dato Elementales / Simples
- Tipos de Dato Primitivos
- Tipos de Dato Estructurados / Complejos

Asignación

- Operación básica para cambiar la ligadura de un valor a un OD
- Efecto colateral:
  - C, Lisp devuelve un valor: OD con una copia del valor asignado
- Especificación
  - Pascal:

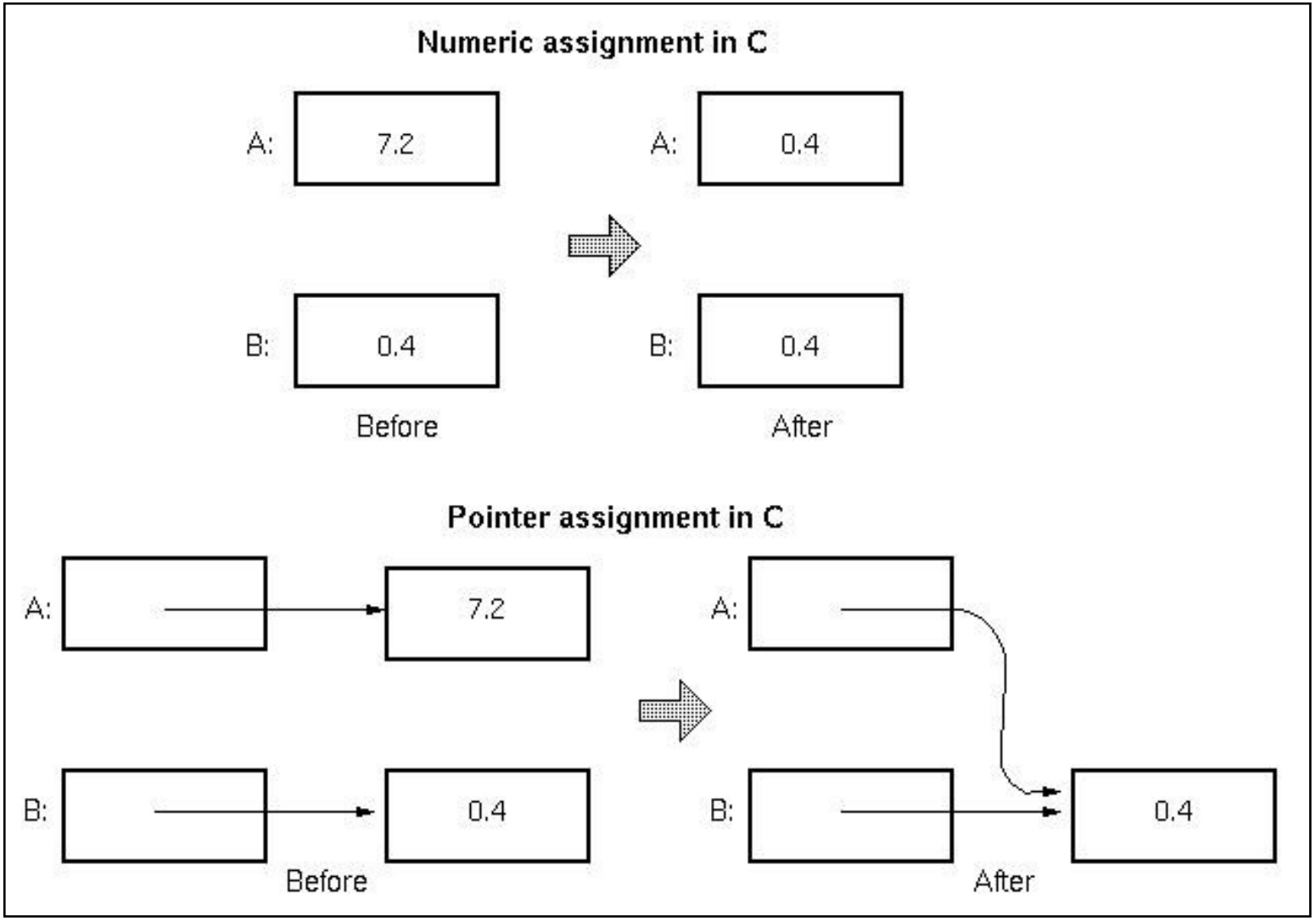
    Asignación(:=):int_1 x int_2→ vacío

  - C:

    Asignación(=):int_1 x int_2 → int

Valor de int_1 sea una copia de ent_2 (ef colateral modificación int_1)

Asignación


Numeric assignment in C
A:
7.2
B:
0.4
Before
A:
0.4
B:
0.4
After
Pointer assignment in C
A:
7.2
B:
0.4
Before
A:
B:
0.4
After

Tipos de Datos Elementales

- Tipos de dato numéricos
  - Enteros
  - Subrangos
  - Reales con punto variable (flotante)
  - Reales con punto fijo
- Enumerados
  - Conjunto de valores simbólicos (us. Ordenados)
- Booleanos
- Caracteres

Tipos de Dato Estructurados

- Objetos de dato estructurados o datos con una estructura:
  - Construcción como agregación de otros objetos de dato, llamados componentes
- Consideraciones:
  - Cómo se establecen los componentes
  - Cómo se relacionan los componentes
  - Cómo se gestiona el almacenamiento
  - Cómo se manipulan las estructuras

Tipos de Dato Estructurados: Especificación

- Número de componentes
- Tipo de cada componente
- Nombres / Funciones para seleccionar un componente individual
- Máximo número de componentes
- Organización de los componentes

Tipos de Dato Estructurados: atributos

- Número de componentes
  - Tamaño Fijo
    - Arreglos, registros, cadena de caracteres
  - Tamaño Variable
    - Pilas, listas, conjuntos, tablas, archivos, cadenas de caracteres
    - Gestión a través de tipo puntero
    - Operaciones de crecimiento y decrecimiento, inserción y borrado

Tipos de Dato Estructurados: atributos

- Tipo de cada componente
  - Homogéneos
    - Arreglos, cadena de caracteres, conjuntos, archivos
  - Heterogéneos
    - Registros, listas

Tipos de Dato Estructurados: atributos

- Nombre / Función de selección de cada componente
  - Subíndice entero o secuencia de subíndices
    - Arreglos
  - Identificador definido por el programador
    - Registros
  - Operaciones
    - Acceso secuencial: pilas, archivos
    - Acceso aleatorio: archivos
  - Operaciones sobre estructura completa

Tipos de Dato Estructurados: atributos

- Organización de los componentes
  - Secuencia lineal (simple)
    - Vectores, arreglos, cadenas de caracteres, pilas, listas, archivos
  - Multidimensionales
    - Arreglos
    - Registros
    - Listas

Tipos de Datos Elementales

- Casi todos los lenguajes de programación proveen un conjunto de tipos de dato primitivos
- Tipo de Dato Primitivo/Simple:
  - No se define en términos de otro tipo de dato
- Algunos tipos de dato primitivos son reflejo del hardware subyacente
- Otros tipos requieren una leve adecuación para su implementación en hardware

Tipo de Dato Numérico

- Entero / Integer
  - Muchas veces como un reflejo exacto del entero propio del hardware.
  - Pueden existir varios subtipos en los lenguajes
    - Hasta 8 tipos de enteros en algunos
    - byte, short, int, long, char

Tipo de Dato Entero

- Entero / Integer
  - Especificación
    - Valores mínimo y máximo (tamaño), ordenados
    - Signo o sin Signo
    - Operaciones
      - Aritméticas: ej. BinOp: integer x integer → integer
      - Relacionales: RelOp: integer x integer → boolean
      - Asignaciones: Asig: ej. integer x integer → void
      - Operaciones a nivel bit: BinOp: integer x integer → integer
  - Implementación
    - Definidas por hardware

Bibliografía

- Pratt, Terrance W., Programming Languages: Design and Implementation, cap 4
- Sebesta, Robert, Concepts of Programming Languages, Cap 6