



Principios de Lenguajes de Programación

Tipo de Datos:

Elementales / Primitivos / Simples

Facultad de Informática
Universidad Nacional del Comahue

Primer Cuatrimestre



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 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.
 - El Mapeo es trivial (en este caso)
 - Pueden existir varios subtipos en los lenguajes
 - Hasta 8 tipos de enteros en algunos
 - Entero con Signo en Java:
 - byte, short, int, long

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



Tipo de Dato Flotante

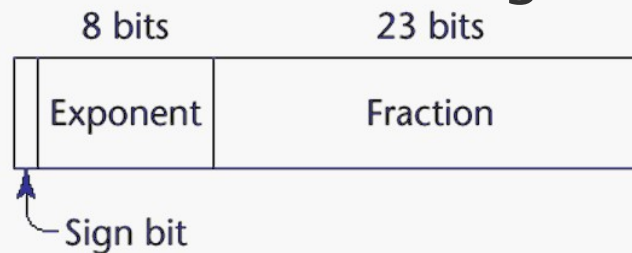
- Modela número reales, pero sólo aproximaciones
 - Incluso 0.1 no puede representarse con exactitud
- Uso científico
 - Más de un tipo: float (4 bytes) double (8 bytes)
- Generalmente es como el hardware subyacente, con representación IEEE Floating-Point Standard 754.

Tipo de Dato Flotante

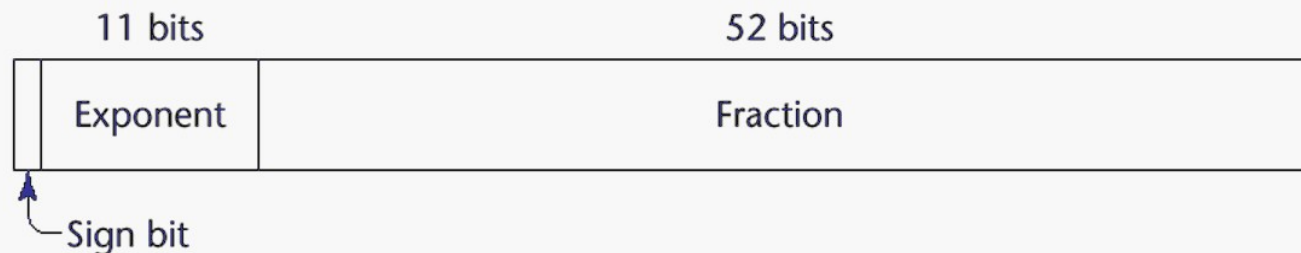
- Especificación:
 - Valor mínimo y máximo
 - Redondeo: control de igualdad puede general fallas
- Implementación
 - Mantisa – Exponente
 - IEEE – 754 (signo, exponente, fracción)
 - Flotante (1, 8, 24) binary32
 - Doble (1, 11, 53) binary64
 - Quádruple (1,15,112) binary128

Tipo de Dato Flotante

- Representación IEEE Floating-Point Standard 754.
 - Real en FORTRAN, float en C
 - Precisión: de la parte fraccionaria (número de bits)
 - Rango: combinación del rango de fracción y exponente



(a)



(b)



Tipo de Dato Punto Fijo

- Especificación:
 - Número real con lugares predefinidos para guardar decimales
- Implementación:
 - Directamente por Hardware o simulada por software



Tipo de Dato Boolean

- Uno de los más simples
- Rango de Valores
 - Dos elementos, 1 para “verdadero” y 0 “falso”
- Implementación
 - Como bits (usando banderas)
 - us. como bytes
- Ventaja:
 - Facilidad de lectura – comprensión del programa



Tipo de Dato Caracter

- Us. almacenado con codificación numérica
- Codificación:
 - Más usada: ASCII
 - Alternativas: Unicode (UCS-2 – 16 bits)
 - Caracteres de casi todos los lenguajes naturales
 - Orig. Java, C# y JavaScript
 - Unicode (UCS-4 – 32 bits)
 - Fortran, desde 2003



Tipo de Dato Cadena (String)

- Valores almacenados como secuencia de caracteres
- Consideraciones de diseño:
 - ¿Es un tipo primitivo o tipo especial de arreglo?
 - ¿Longitud: estática o dinámica?



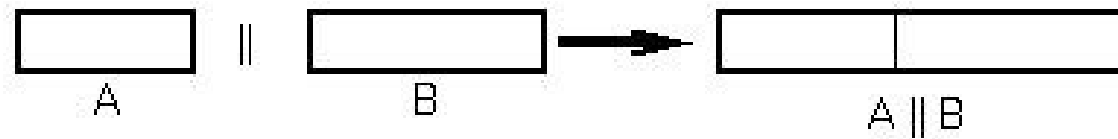
Tipo de Dato Cadena (String)

- Operaciones:
 - Asignación y copia
 - Comparación (=, >, etc.)
 - Contatenación
 - Referencia a subcadena
 - Subíndices
 - Concordancia de patrones (pattern matching)

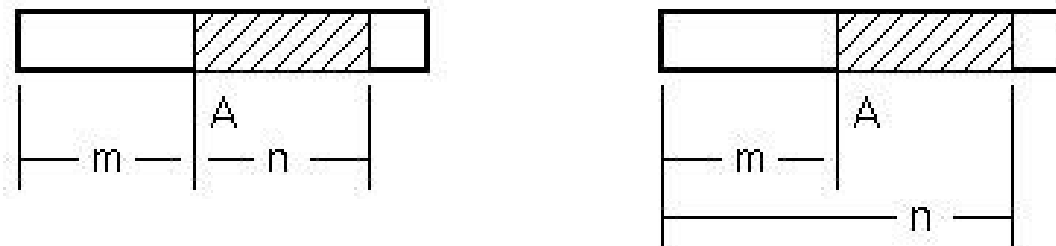
Tipo de Dato Cadena (String)

OPERATIONS

Concatenation: $A \parallel B$



Substring: $\text{substring}(A, m, n)$



Bit strings -- same, but at bit-level

Tipo de Dato Cadena (String)

- C / C++
 - No es primitivo
 - Usa arreglos tipo “char” y funciones de librería con operaciones
- SNOBOL4 (lenguaje de manejo de cadenas)
 - Primitivo
 - Muchas operaciones, concordancia de patrones complejo
- Fortran / Python
 - Primitivo con asignación y otras operaciones
- Java
 - Primitivo via clase String
- Perl, JavaScript, Ruby, y PHP
 - Concordancia de patrones por expresiones regulares

Tipo de Dato Cadena (String)

- Opciones de Longitud
 - Estático: COBOL, Clase `String` Java
 - Longitud dinámica limitada: C / C++
 - Caracter especial para indicar fin de cadena
 - No se mantiene longitud en descriptor
 - Dinámico (sin máximo): SNOBOL4, Perl, JavaScript
 - Ada: las tres opciones



Tipo de Dato Primitivo: Cadena (String)

- Evaluación:
 - Ayuda a la facilidad de escritura
 - Como tipo primitivo de longitud estática
 - Simples y baratos ¿porqué no tenerlos?
 - Longitud dinámica:
 - Útil, pero ¿vale el costo?



Tipo de Dato Primitivo: Cadena (String)

- Especificación:
 - Longitud fija:
 - Guarda lugar en tiempo de compilación
 - Cadenas más largas se truncan
 - Longitud variable con límite declarado:
 - Guarda lugar en tiempo de compilación
 - Límite superior para la longitud actual
 - Cadenas más largas que máximo se truncan
 - Longitud dinámica:
 - Guarda alocaión en tiempo de ejecución
 - Cadenas de cualquier longitud

Tipo de Dato Primitivo: Cadena (String)

- Descriptor de tiempo de compilación para cadenas estáticas
- Descriptor de tiempo de ejecución para cadenas dinámicas limitadas

Static string
Length
Address

Limited dynamic string
Maximum length
Current length
Address



Tipos ordinales definidos por usuario

- Tipo ordinal:
 - El rango de los valores posibles puede asociarse con un conjunto de enteros positivos
- Ejemplos de tipo ordinal en Java
 - `integer`
 - `char`
 - `boolean`

Tipo de Dato Primitivo: enumerados

- Todos los valores posibles, con constantes con nombre, como parte de la definición
- Ejemplo C#

```
enum dias {mon, tue, wed, thu, fri, sat, sun};
```
- Consideraciones de Diseño:
 - Constante de enumerado, ¿puede aparecer en más de una definición de tipo? Entonces ¿cómo se controla la ocurrencia de esta constante?
 - Los valores, ¿se coartan a enteros?
 - Otro tipo, ¿se coarta a un tipo enumerado?



Tipo de Dato Primitivo: enumerados

- Evaluación:
 - Ayuda a la facilidad de lectura
 - No requiere codificar un color como un número
 - Ayuda a la confiabilidad, (el compilador puede controlar):
 - Operaciones (no permite que se sumen los colores)
 - Variables enumeradas no se le asignan valores fuera de rango
 - Ada, C#, y Java 5.0: buen soporte
 - C++: coerción a tipos enteros

Tipo de Dato Primitivo: subrango

- Subsecuencia contigua ordenada de un tipo ordinal:
 - Ejemplo: 12..18 subrango de tipo entero
- Diseño de Ada:

```
type Days is (mon, tue, wed, thu, fri, sat, sun);
```

```
subtype Weekdays is Days range mon..fri;
```

```
subtype Index is Integer range 1..100;
```

```
Day1: Days;
```

```
Day2: Weekday;
```

```
Day2 := Day1;
```



Tipo de Dato Primitivo: subrango

- Evaluación:
 - Ayuda a la facilidad de lectura
 - Las variables de subrangos almacenan sólo un cierto rango de valores
 - Confiabilidad:
 - Valor fuera de rango, se detecta como error

Tipo de Dato Primitivo: subrango

- Evaluación:
 - Confiabilidad: C++ mejor que C
 - Las variables numéricas se asignan a tipo enumerado si son “casteadas” al tipo de variable
 - Valores numéricos se verificar para controlar el rango
 - Sólo sirve para consecutivos, problemas en

```
enum colors {red = 1, blue = 100, green = 10000}
```



Tipo de Dato Primitivo: punteros

- Simple en algunos lenguajes
- Especificación:
 - Referencia objetos de datos de un *único* tipo: C, Pascal, Ada, etc.
 - Referencia objetos de datos de cualquier tipo: Smalltalk, etc.
 - Punteros como objetos de datos manipulados por el programa. C, C++, etc.
 - Punteros como estructuras ocultas, gestionadas por la implementación del lenguaje: Java

Tipo de Dato Primitivo: punteros

