Lenguajes de Programación

Entidades, atributos y ligaduras

Ma. Laura Cobo

Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
2018

Entidades

Nombres o identificadores:

cadena de caracteres utilizado para identificar entradas en un programa

Consideraciones de diseño:

- ¿tienen longitud máxima?
- ¿es sensible a mayúsculas y minúsculas?
- ¿se permite el uso de conectores?
- ¿hay palabras especiales (claves o reservadas)?

2

Diseño de nombres

Longitud

si es muy restringida afecta la legibilidad del programa. Ej:

- Fortran: longitud máxima 6
- Cobol: longitud máxima 30
- Ada y Java: sin limites, todos los caracteres son significativos
- C++: sin limites, pero los implementadores generalmente imponen uno

Uso de Conectores: muchos lenguajes no los permiten

Sensibilidad a mayúsculas y minúsculas: interfiere seriamente con la facilidad de lectura (readabiliy)

3

Diseño de nombres

Palabras especiales

- Palabras clave: aumentan la facilidad de lectura, ya que funcionan como separadores.

  Una palabra clave, es un identificador que solo es especial (en interpretación) en ciertos contextos. Por ejemplo en Fortran:

  Real VarName (Real *es una palabra clave en este contexto)*
  Real = 3.4 (*Real es aquí el nombre de una variable)*

- Palabras reservadas: es una palabra clave que no puede redefinirse por el usuario. Tienen su semántica claramente definida.

- Palabras predefinidas: conjunto d
  predeterminado, pero que puede ser modificado por el programador.

El abuso de este tipo de palabras en un lenguaje perjudica su aprendizaje y evolución

4

Atributos de una variable

- Nombre: no todas las variables poseen uno
- Dirección: la dirección de memoria a la que está asociada
  - ✓ una misma variable puede estar asociada a locaciones de memoria diferentes durante la ejecución del programa, y en diferentes lugares del programa.
  - ✓ Alsiasing: dos nombres de variables diferentes pueden utilizarse para acceder a la misma locación de memoria al mismo tiempo

    El aliasing perjudica la legibilidad de los programas, ya que el programador que lee el programa debe tener presente todos los alias de una locación.

    Los alias se pueden crear a través del uso de punteros, variables por referencia.

5

El concepto de ligadura

Una *ligadura* es la asociación entre dos cosas.

Por ejemplo: se da entre un nombre y la cosa nombrada por él.

El *tiempo de ligadura* es el momento en cuál se crea la liagadura o el momento en el cuál se toma una decisión de implementación.

Hay diferentes momentos a los cuales la decisiones pueden estar vinculadas.

6

Tiempo de ligadura

Diseño del lenguaje: muchos de los aspectos de semántica se establecen en este momento (flujo de control, conjunto de tipos)

Implementación del lenguaje: la mayoría de los lenguajes dejan detalles a la discreción del implementador (el numero de bits para la representación de un tipo, tamaño del heap y pila, manejo de excepciones en tiempo de ejecución)

Escritura del programa: algoritmos, estructuras de datos, nombres

Compilación: mapeo de los constructores alto nivel a código maquina o datos estáticos a memoria

7

Tiempo de ligadura

Linkeo: la compilación separada hace que algunas ligaduras no puedan resolver en compilación.

Carga: hace referencia al momento en el que el sistema operativo carga en memoria el programa. En el linkeo se establecen direcciones virtuales, en la carga se resulven las direcciones físicas.

Ejecución: cubre las ligaduras que tienen lugar desde el comienzo al fin de la ejecución, por ejemplo el valor de las variables

8

Tiempo de ligadura

Tiempos de ligadura

- Ejecución
- Carga
- Linkeo
- Compilación
- Escritura del programa
- Implementación del lenguaje
- Diseño del lenguaje

Ligadura dinámica

Ligadura estática

La ligadura temprana esta asociada a mayor **eficiencia**, mientras que la mas demorada a mayor **flexibilidad**

9

Variables

Una variable es una abstracción de una celda de memoria.

Los atributos que caracterizan a una variable son:

- Nombre
- Dirección
- Valor
- Tipo
- Tiempo de vida
- Alcance

Para el paradigma imperativo

10

Constantes nombradas

Una constante nombrada es una variable que se liga a un valor sólo cuando se realiza la ligadura a su correspondiente locación de memoria.

Inicialización de Variables

La ligadura de una variable a su valor en el momento que se liga al medio de almacenamiento se llama inicialización

Generalmente toma lugar a través de una asignación.

Ejemplo (Java):

```
int contador = 0;
```

11

Atributos de una variable

- Valor: es el contenido de la celda o celdas asociadas a la variable.
- Tipo: determina el rango de valores para la variable y las operaciones que están definidas para los valores de ese tipo.
  - ✓ Ligadura del tipo: antes de que una variable pueda ser utilizada debe estar ligada a un tipo.

    Es relevante entonces, decidir dos aspectos:

    1. Como se especifica el tipo
    2. Cuando la ligadura toma lugar

12

Ligadura del tipo a una variable

- Ligadura estática: el tipo se determina a través de una declaración explícita o implícita
- Ligadura dinámica: en algunos lenguajes se determina el tipo a través de las sentencias de asignación. Por ejemplo en PHP o JavaScript.
  - Ventaja: flexibilidad
  - Desventajas: altos costos (chequeo de tipos dinámico e interpretación). La detección de errores de tipos en compilación es muy difícil.

Ejemplo en JavaScript

Lista = [2, 4.33, 3.1416, 8]

Lista = 7,14

13

Declaración explícita-implícita

- Declaración explícita: sentencia del programa en la cual se declara el tipo de la variable
- Declaración implícita: es un mecanismo por default para especificar el tipo de una variable (la primera aparición de la variable en el programa)
  - Ventaja: facilidad de escritura
  - Desventajas: confiabilidad

  Ejemplos de lenguajes con declaraciones implícitas: Fortran, PL/I, Basic y Pearl
- Inferencia de tipo: mecanismo utilizado por ML, el tipo de la expresión es determinado por el contexto

Ejemplo en ML

Fun Mult10(x) = 10 * x **Fun** cuadrado(x): real = x * x

14

Chequeo de tipos

- Es una actividad que permite asegurar que una operación y sus operandos son de tipos compatibles (subprograma = operación, parámetros = operandos)
- Un tipo es compatible si es legal para la operación o puede convertirse implícitamente a un tipo legal (coersión)
- Un error de tipos sucede cuando se aplica una operación a un operando de tipo inapropiado.
- Si todas las ligaduras de tipo son estáticas, entonces el chequeo de tipos se puede hacer casi completamente estático. Si la ligadura es dinámica el chequeo debería ser dinámico.
- Un lenguaje de programación es fuertemente tipado, si los errores de tipos **siempre** son detectados

15

Compatibilidad de tipos

La idea de compatibilidad se definió por necesidad para el chequeo de tipos.

Hay dos tipos de reglas de compatiblidad para variables de tipos estructurados (no escalares).

Influencian el diseño de los tipos de datos y las operaciones provistas para ese tipo.
Lo mas importante sobre compatibilidad, es saber si un tipo puede asignar su valor al otro

Compatibilidad por nombre

Compatibilidad por estructura

16

Compatibilidad por nombre

Dos variables son compatibles si:

1. Están en la misma declaración
2. Están en declaraciones que usan el mismo tipo (mismo nombre de tipo)

Este esquema de compatibilidad, es fácil de implementar pero demasiado restrictivo

1. Los subrangos no son compatibles con su tipo base
2. Los parámetros actuales deben ser los mismos que los formales

Compatibilidad por estructura

Dos variables son compatibles por estructura si:

Las dos variables tienen tipos que tiene estructuras idénticas

Este esquema de compatibilidad, es más flexible pero mas difícil de implementar

18

Tiempo de vida y manejo de almacenamiento

- Es importante distinguir entre nombres y los objetos a los que hacen referencia
- Hay varios eventos vinculados a esta diferencia:
  - ✓ Creación de objetos
  - ✓ Creación de ligaduras
  - ✓ Referencias a variables, subrutinas, tipos, etc (todos usan ligaduras)
  - ✓ Desactivación y reactivación de ligaduras que pueden estar temporalmente inutilizables
  - ✓ Destrucción de ligaduras
  - ✓ Destrucción de objetos

El **tiempo de vida**, es el período de tiempo entre la creación y destrucción de la ligadura entre un nombre y un objeto

19

Tiempo de vida y manejo de almacenamiento

El **tiempo de vida**, es el período de tiempo entre la creación y destrucción de la ligadura entre un nombre y un objeto

El tiempo de vida del objeto corresponde al mecanismo de asignación de memoria

- Objetos Estáticos: los objetos están asociados a direcciones absolutas de memoria. Esta asociación se retiene/mantiene durante toda la ejecución.
- Objetos Pila: se asignan o desasignan (allocate o deallocate) de manera FIFO, usualmente con las llamadas a rutinas y retornos
- Heap: se asignan y desasignan en tiempos arbitrarios. Requiere algoritmos de manejo de almacenamiento mas generales

20

Tiempo de vida: Categoría de variables

- Estáticas: la variable queda ligada a la celda de memoria antes de la ejecución y permanece ligada a la misma durante toda la ejecución. Por ejemplo: todas las variables Fortran y las variables C declaradas static.

  Ventajas: eficiencia (acceso directo), soporte para subprogramas sensibles a la historia.

  Desventajas: falta de flexibilidad. En un contexto con sólo variables estáticas es imposible definir recursión

21

Tiempo de vida: Categoría de variables

- Dinámicas en Pila: la ligadura al almacenamiento se realiza cuando se elabora la sentencia de declaración.

Ventajas:

Admite recursión

Conserva el almacenamiento

Desventajas:

Overhead a la hora de alocar y desalocar memoria

No se pude tener sensibilidad a la historia.

Referencias ineficientes(direccionamiento indirecto)

22

Tiempo de vida: Categoría de variables

- Dinámicas con heap explícito: se aloca y dealoca memoria a través de directivas explícitas, realizadas por el programador y que tienen efecto durante la ejecución

  Las variables son referenciadas, a través de punteros o referencias. Por ejemplos las variables dinámicas en C++ y todos los objetos en Java

  Ventajas:

  Provee manejo dinámico del medio de almacenamiento

  Desventajas:

  Ineficiente y poco confiable

Tiempo de vida: Categoría de variables

- Dinámicas con heap implícito: la alocación y dealocación se produce a través de las sentencias de asignación

  Ejemplos: todas las variables en APL; los strings y arrays en Pearl y JavaScript

  Ventajas:

  flexibilidad

  Desventajas:

  Ineficiente, ya que todos los atributos son dinámicos

  Pérdida de detección de errores

24

Atributos de una variable (continuación)

- Alcance: es la región textual de un programa en el cual una ligadura esta activa

> Una variable tiene una ligadura activa para una sentencia si puede ser referenciada en ella

El alcance de una ligadura puede determinarse:

- Estáticamente
- Dinámicamente

25

Alcance

Regla de alcance: Determinan como referencias de variables declaradas fuera del subprograma en ejecución o bloque son asociadas con sus declaraciones y por lo tanto con sus atributos

Alcance estático:

en un lenguaje con reglar de alcance estático o léxico la ligadura se determina en tiempo de compilación a partir de la examinación del texto del programa.

Habitualmente la ligadura tiene lugar “matcheando” la declaración cuyo bloque esta mas cercano a ese punto del programa.

26

Alcance

Alcance estático:

en un lenguaje con reglar de alcance estático o léxico la ligadura se determina en tiempo de compilación a partir de la examinación del texto del programa.

Habitualmente la ligadura tiene lugar “matcheando” la declaración cuyo bloque esta mas cercano a ese punto del programa.

Formas conocidas en lenguajes:

- local – global
- Bloques comunes e importación
- Bloques anidados
- Orden de declaración

27

Alcance

Alcance Dinámico: la ligadura se resuelve siguiendo la secuencia de llamados, no en la relación del texto del programa. De ahí que solo pueda determinarse en ejecución.

28

Ambiente de referenciamiento

El ambiente de referenciamiento de una sentencia es la colección de nombres que son visibles en la sentencia.

- En lenguajes con reglas de ambiente estático: el ambiente esta conformado por todos los nombres locales mas los nombres de todos los ambientes que lo contienen
- En lenguajes con reglas de ambiente dinámico: el ambiente esta conformado nuevamente por todos los nombres locales, mas los nombre visibles de todos los subprogramas activos.

Un subprograma está activo, si su ejecución ha comenzado pero aún no ha finalizado

29

Resumiendo ….

El alcance de una entidad es la región de texto del programa en el cuál una determinada entidad es visible (su ligadura está activa).

El ambiente de referenciamiento es la colección o conjunto de entidades visibles (con ligaduras activas) en una determinada sentencia.

30

El significado de los nombres dentro de su alcance

¿Qué sucede su el mapeo entre nombre y objetos no es uno-a-uno?

- alias: el mismo objeto de dato está ligado a mas de un nombre en el alcance.
- Sobrecarga de operaciones

31