Programming Languages

Lecture 6: Bindings

Benjamin J. Keller

Department of Computer Science, Virginia Tech

Binding Time

- Attributes of parts of programs must be "bound" to object before or during computation.
- A binding fixes a value or other property of an object (from a set of possible values)
- Time at which choice for binding occurs is called binding time.
  - Dynamic binding — at execution
  - Static binding — at translation, language implementation, or language definition

Dynamic Binding

- At entry to block or subprogram
  - Bind actual to formal parameter
  - Determine location of local variable
- At arbitrary times in program — bind values to variables via assignment

Static Binding

- At translation
  - Determined by programmer — bind type to variable name, values to constants
  - Determined by translator — bind global variable to location (at load time), bind source program to object program representation
- At implementation
  - Bind values to representation in computer
  - Bind operations and statements to semantics (if not uniform may lead to different results with different implmentations)

Static Binding (cont)

- At language definition
  - Structure of language
  - Built-in and definable types
  - Notation for values

Binding Time Examples

1. When is meaning of "`+`" bound to its meaning in "`x + 10`"?
   - Could be at language definition, implementation, or at translation
   - May also be execution time — could depend on type of `x` determined at run-time
2. Difference between reserved and keywords has to do with binding time
   - Both bound at language definition, but reserved word binding can't be changed
   - Ex. "`DO`" is reserved word in Pascal, but not FORTRAN (can write `DO = 10`)
   - Ex. "Integer" may be redefined in Pascal, but not FORTRAN or Ada.

Late vs. Early Binding Time

- Many language design decisions relate to binding time
  - Late — more flexible
  - Early — more efficient
- Ex. More efficient to bind "+" at translation than execution
- Early — supports compilation, late — supports interpretation
- Programming choices may delay binding time
- Ex. recursion forces delay in binding time of local variables to locations (FORTRAN allows choice: static allocation vs stack-based allocation)
- Generally considered useful to bind ASAP

Managing Bindings

- Bindings stored both at compile and at run-time.
- Compilation
  - Declarations stored in Symbol table ($\mathit{Names} \rightarrow \mathit{Attributes}$)
  - Most bindings used only in the compilation process
- Execution
  - Run-time environment keeps track of meanings of names ($\mathit{Names} \rightarrow \mathit{Locations}$)
  - Contents of locations stored in memory (also called the *state*) ($\mathit{Locations} \rightarrow \mathit{Values}$)
- An interpreter keeps all 3 kinds of bindings together in one environment