CS450

Structure of Higher Level Languages

Lecture 31: Dynamic binding

Tiago Cogumbreiro

Today we will learn...

- Revisit dynamic binding
- Dynamic binding to control globals
- Dynamic binding to control testing

Dynamic scoping in Racket

`parameterize`

Static versus dynamic scoping

Static Scoping

> **Static binding:** variables are captured at creation time

Dynamic Scoping

> **Dynamic binding:** variables depends on the calling context

Why dynamic scoping?

1. A controlled way to represent global variables
2. A technique to make code testable

Dynamic scoping example

Dynamic scoping In Racket

Pseudo-Racket dynamic scoping

- Function returns a reference to a dynamically scoped memory-cell
- Calling a parameter without parameter returns the contents of the memory-cell
- Use to overwrite the memory-cell

Dynamic binding

Globals

Dynamic binding: controlled globals

We can define different globals in different contexts.

Racket uses parameters to allow extending the behavior of many features:

- command line parameters
- standard output stream (known as a port)
- formatting options (eg, default implementation to print structures)

Dynamic binding

Testing

Dynamic binding: making code testable

Consider an excerpt of Homework 5. We would like to be able to test each function independently. How?

Dynamic binding: making code testable

- In Homework 4, we added a function parameter to test independently from .
- This extra function parameter was confusing to some students.
- This choice made the function interface more verbose than needed.
- More arguments, more chance of mistakes! Do we call or ?

How can we use dynamic binding

to improve the testing design of `r:eval`?

Dynamic binding: making code testable

- Create a parameter per global function that you want to make testable
- Internal calls should target the ***parameter*** and not the global variable

Before

Dynamic binding: making code testable

- Create a parameter per global function that you want to make testable
- Internal calls should target the ***parameter*** and not the global variable

Before

After

Dynamic binding: making code testable

> Consider an excerpt of Homework 5. We would like to be able to test each function independently. How?

Dynamic binding: making code testable

- Usage example:

We can test eval-term without implementing eval-exp!

This testing technique is known as ***mocking***.

Delimited dynamic binding

ICFP 2006

- Source: ________________________________
- Resources: ____