Mutable Variables

CS 350

Dr. Joseph Eremondi

Last updated: August 1, 2024

Outline

Overview

- Learning goals

Overview

- Learning goals
    - To interpret a language where variables can change values (mutate)

Overview

- Learning goals
  - To interpret a language where variables can change values (mutate)
  - To understand the design choices around functions in such a language

Overview

- Learning goals
  - To interpret a language where variables can change values (mutate)
  - To understand the design choices around functions in such a language
- Key concepts

Overview

- Learning goals
  - To interpret a language where variables can change values (mutate)
  - To understand the design choices around functions in such a language
- Key concepts
  - Pass-by-reference vs. Pass-by-value

Identifiers vs Variables

- Until now, a variable denoted a *value*

Identifiers vs Variables

- Until now, a variable denoted a *value*
  - In a given environment

Identifiers vs Variables

- Until now, a variable denoted a *value*
    - In a given environment
- They didn’t really ever vary

Identifiers vs Variables

- Until now, a variable denoted a *value*
    - In a given environment
- They didn’t really ever vary
    - Except between different function calls

Identifiers vs Variables

- Until now, a variable denoted a *value*
  - In a given environment
- They didn’t really ever vary
  - Except between different function calls
- To allow variables values to change, we can make one simple change:

Identifiers vs Variables

- Until now, a variable denoted a *value*
  - In a given environment
- They didn’t really ever vary
  - Except between different function calls
- To allow variables values to change, we can make one simple change:
  - **Keep locations instead of values in the environment**

Identifiers vs Variables

- Until now, a variable denoted a *value*
  - In a given environment
- They didn't really ever vary
  - Except between different function calls
- To allow variables values to change, we can make one simple change:
  - **Keep locations instead of values in the environment**
  - Then each variable refers to a single store location, whose value can change

Pass-by-value

Pass-by-reference