Discussion 14

Mutability :O

Kenneth Fang (kwf37), Newton Ni (cn279)

March 19, 2019

Agenda

1. Review of Mutable Constructs
2. Exercises
3. Rec 14

Mutable Fields

- A record can have a mutable field by using the `mutable` keyword.
- Ex:

```
type mutable_student = {
name: string,
mutable age: int
}
```

Mutable Fields

Try this in utop:

```
type mutable_student = {
  name: string,
  mutable age: int
}
```

- `let kenneth = {name="kenneth";age=20}`
- `kenneth.age <- 5`
- What type did the last command return?
- What happens if you type `kenneth` now?
- Try running `kenneth.name <- "newton"`

Refs

- Refs are just records with a single mutable field:
- From lecture:

```
type 'a ref = { mutable contents: 'a }
let ref x = { contents = x }
let ( ! ) r = r.contents
let ( := ) r newval = r.contents <- newval
```

- How can we use these?

Refs

Time for more utop:

- `let x = ref 0`
- `x := !x + 1`
- `x`
- `let y = ref 1`
- `x = y (* What does this evaluate to? *)`
- `x == y (* How about this? *)`

Physical Equality

Time for more utop:

- (==) is the physical equality operator

Physical Equality

Time for more utop:

- (==) is the physical equality operator
- It checks if two mutable values point to the same location in memory

Physical Equality

Time for more utop:

- `(==)` is the physical equality operator
- It checks if two mutable values point to the same location in memory
- Its complement is `(!=)` for physically not equal

Physical Equality

Time for more utop:

- (==) is the physical equality operator
- It checks if two mutable values point to the same location in memory
- Its complement is (!=) for physically not equal
- The regular equality operators (=) and (<>) check if all the contents of each mutable field are the same

Physical Equality

Time for more utop:

- (==) is the physical equality operator
- It checks if two mutable values point to the same location in memory
- Its complement is (!=) for physically not equal
- The regular equality operators (=) and (<>) check if all the contents of each mutable field are the same

Physical Equality

Some questions: (discuss with your neighbors and raise your hand when you have an idea)

- Is it possible for (=) to return true and (==) to return false?
- Is it possible for (=) to return false and (==) to return true?

Physical Equality/Aliasing

Try out the following:

- `let x = ref "wowie"`
- `let z = x`
- `x == z (* What does this evaluate to? *)`
- `z := "2b || !2b"`
- `x`
- `z`

Aliasing

- Aliasing is when two pointers point to the same location in memory

Aliasing

- Aliasing is when two pointers point to the same location in memory
- It should be familiar from programming in Java and other imperative languages

Aliasing

- Aliasing is when two pointers point to the same location in memory
- It should be familiar from programming in Java and other imperative languages
- It can be hard to keep track of!

Exercises

- As usual, pull from the repo and check out the exercises

Rec 14