CSC 347 - Concepts of Programming Languages

Iteration: Loops and Recursion

Instructor: Stefan Mitsch

Learning Objectives

❓ How to implement iteration?

- Understand loops vs. recursion

Loops by Recursion

- Loop forever printing `hello` (press Control-C to quit)

```
while true do println("Hello")
```

- Recursive implementation

```
def loop() = println("Hello"); loop()
```

- With a loop counter

```
def loop(int n) = println(s"Hello $n"); loop(n+1)
```

Translating Loop to Recursion

Loop (mutable data)

```
def factorial (n:Int) : Int =
  val result = 1
  var m = n
  while m > 1 do
    result = result ∗ m
    m = m − 1
  result
```

- **Recursive (mutable)**

```
def factorial (n:Int) : Int =
  var result = 1
  var m = n
  def loop () : Unit =
    if m > 1 then
      result = result∗m
      m = m−1
      loop()
  loop()
  result
```

- **Recursive (mutable)**

```
def factorial (n:Int) : Int =
  var result = 1
  def loop (m:Int) : Unit =
    if m > 1 then
      result = result∗m
      loop(m−1)
  loop(n)
  result
```

- **Recursive (immutable)**

```
def factorial (n:Int) : Int =
  def loop (m:Int) : Int =
    if m > 1 then m ∗ loop(m−1)
    else 1
  loop(n)
```

- **Tail-recursive**

```
def factorial (n:Int) : Int =
  def loop (m:Int, result:Int) : Int =
    if m > 1 then loop(m−1, m∗result)
    else result
  loop(n,1)
```

Summary

- Iteration with loops often uses mutable data
- Iteration with recursion often uses immutable data
- Can translate between loops and recursion