

# Logic and Computation – CS 2800

## Fall 2019

### Lecture 22

### Measure functions: more examples

Stavros Tripakis



**Northeastern University**  
**Khoury College of**  
**Computer Sciences**

# Outline

- More about measure functions

# Measure functions: more examples

# Examples from lab 08

Measure functions?  
Proof obligations?

```
(definec app?-t2 (x :tl y :tl acc :tl) :tl
  (if (endp x)
      (if (endp y)
          acc
          (app?-t2 x (rest y) (cons (first y) acc)))
      (app?-t2 (rest x) y (cons (first x) acc))))
```

```
(definec app?-t3 (x :tl y :tl acc :tl) :tl
  (cond ((and (endp x) (endp y)) acc)
        ((endp x) (app?-t3 x (rest y) (cons (first y) acc)))
        ((endp y) (app?-t3 (rest x) y (cons (first x) acc)))
        (t (app?-t3 x nil (app?-t3 nil y acc)))))
```

```
(definec app?-t4 (x :tl y :tl acc :tl) :tl
  (cond ((and (endp x) (endp y)) acc)
        ((endp x) (app?-t4 x (rest y) (cons (first y) acc)))
        ((endp y) (app?-t4 y x acc))
        (t (app?-t4 x nil (app?-t4 acc nil y)))))
```

# Quiz

Do these functions always terminate?

```
(definec drop-last (x :tl) :tl
  (if (endp (rest x))
      nil
      (cons (first x) (drop-last (rest x)))))
```

```
(definec prefixes (X :tl) :tl
  (if (endp X)
      ' ( () )
      (cons X (prefixes (drop-last X)))))
```

# Next time

- Undecidability