# Comp 311
# Principles of Programming Languages
# Lecture 10
# The Semantics of Recursion

Corky Cartwright

September 15, 2010

# Key Intuitions

- Computation is incremental not monolithic
- Slogan: general computation is successive approximation (typically in response to successive demand for more information).

# Key Mathematical Concepts

Domains:

- (weak) partial order
- chain
- chain-complete
- complete partial order (**cpo**)
- “home-plate” **cpo**
- consistently-complete
- bot $_{(\perp)}$
- flat domains

# Key Mathematical Concepts

Computable functions:

- monotonic
- continuous
- strict
- flat domain
- “home-plate” **cpo**
- consistently-complete
- **bot** (⊥)

# Examples

Domains

- flat domains
- strict function spaces on flat domains
- lazy trees of boolean (of D where D is flat)
- factorial functional

# A Bigger Challenge

Assume that we want to write LC in a purely functional language without a recursive binding construct (say functional Scheme without `define` and `letrec`)?

- Key problem: must expand `letrec` into `lambda`
- No simple solution to this problem. We need to invoke syntactic magic or develop some sophisticated mathematical machinery.