

Comp 411  
Principles of Programming Languages  
Lecture 10  
The Semantics of Recursion

Corky Cartwright  
February 5, 2014

# 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
- bottom ( $\perp$ )
- flat domains

# Key Mathematical Concepts

Computable functions:

- monotonic (universal)
- continuous (universal)
- strict (typical)

# 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.