Tail Calls and Continuations
(Blackboard Lecture)

CS 4447 / CS 9545

Stephen M. Watt

Tail Calls

- Usual stack layout for a normal function call.
  - Caller-saved registers.
  - Space for function result [may be elsewhere].
  - Space for parameters.
  - Return instruction pointer.
  - ----------------------------------
  - Callee-saved registers.
  - Display.
  - Local and temporary variable area.
  - ----------------------------------
  - Evaluation stack.

Tail Calls

- For a tail-call, we want to
  - Restore the callee-saved registers
  - Pop the current function's parameters
  - Push the new function's parameters
  - Establish the new-function's call frame (callee-saved regs,...)
- Problem:
  New parameter list might be longer, overwriting
  the return address.

- Solution:
  Put the return address *before* the parameters.

  (Alternatively pop the return address into a register and
  push it back after pushing the new parameters.)

Calling Continuations

- Explanation of what continuations are.
- Explanation of trade-offs of different approaches:
  - Whole stack snapshot.
  - Heap allocated stack frames.
  - Chunked stack.