

CSSE 304 Day 35 – call/cc in the interpreter. imperative form.

A place for your notes about the adding call/cc to the interpreted language. `(call/cc receiver)` → `(receiver current-continuation)`

The imperative form slides are mostly code. The rest of this document is a place where you can write down things that I say or things that you observe about that code.

1. Original `reverse*` and `append` code.

2. `cps-version`.

3. `cps-version` with tracing.

4. The trace itself.

5. CPS with data-structures continuations.
6. Why the trace is better now?
7. What are the conditions for code to be in imperative form? Why can I say that recursive calls are then like GOTOs?
8. Notes on the imperative form version.
9. Notes on the “details of transformation” slide.
10. Why doesn’t “normal” tracing help us now?
11. How does `trace-it` work?
12. Comments on the new trace that is produced by `trace-it`: