By Zhiqiang Ren
aren@cs.bu.edu
Content

- How to write lambda
- Closure revisited
- Continuation
- Mid-term 1
How to write lambda

- function (closure) without name
- lam (arg1, arg2, ..., argn) => retval
- lam (arg1: type1, ..., argn: typen) => retval: rettype
- recursive lambda expression?
- all the lambda can be replaced by fun
How to write lambda

- \( \text{fun}\{a,b:t@ype\} A (f:\ (a, a) \rightarrow \text{clorefi}\ b): a \rightarrow \text{clorefi}\ b \)

- \( \text{fun}\{a,b,c:t@ype\} B (f:\ (a, b) \rightarrow \text{clorefi}\ c): (b, a) \rightarrow \text{clorefi}\ c \)

- \( \text{fun}\{a,b,c:t@ype\} \)
  - \( \text{compose} (f:\ a \rightarrow \text{clorefi}\ b, g:\ b \rightarrow \text{clorefi}\ c): a \rightarrow \text{clorefi}\ c \)

- \( \text{fun}\{a,b,c:t@ype\} \text{CP} \)
  - \( (f:\ a \rightarrow \text{clorefi}\ b, g:\ (a \rightarrow \text{clorefi}\ c) \rightarrow \text{clorefi}\ c, h:\ b \rightarrow \text{clorefi}\ c): c \)
Closure revisit

- What is function?
- How to determine the result of the function?
- closure = function with extra arguments
Continuation Style

- Call the previous continuation only once at the end!
Working with ATS

- Do the type checking from time to time.
- Using the makefile.