How to debug OCaml?

CMSC330 Fall 2016
Dr. Anwar Mamat
How to debug OCaml code?

• toplevel
• print
• #trace
• camldebug
• emacs
toplevel

• List.sort_uniq;

• - : ('a -> 'a -> int) -> 'a list -> 'a list = <fun>
let rec member x = function
    []->false
    | h::t->let _= print_string "hd:";print_int h; print_string "\n" in
      (x=h)|| member x t;;

member 4 [1;2;4;5];;
hd:1
hd:2
hd:4
- : bool = true
let string_of_int_list lst =  
    List.fold_right (fun x a->(string_of_int x)^","^a) lst "";;
let prt_int_list lst = print_string (string_of_int_list lst);;
let rec member x = function
    []->false
    |h::t->let _= print_string "tail:";prt_int_list t; print_string "\n" in
    (x=h) || member x t;;
member 4 [1;2;4;5];;
member 4 [1;2;4;5];;
tail:2,4,5,
tail:4,5,
tail:5,
- : bool = true
let add x y = x+y;;
    val add : int -> int -> int = <fun>
#trace add;;
add is now traced.
# add 10 20;;
add  <--  10
add  -->  <fun>
add*  <--  20
add*  -->  30
- : int = 30
let rec member x = function
    []->false
    |h::t->(x=h)| | member x t;;
val member : 'a -> 'a list -> bool = <fun>

#trace member;;
let rec member (x:int) = function
    []->false
    |h::t->(x=h)| | member x t;;
val member : int -> int list -> bool
#trace member;;
#trace

let rec member (x:int) = function
    []->false
    |h::t->(x=h)| | member x t;;
val member : int -> int list -> bool
#trace member;;
let rec member (x:int) = function
    [] -> false
  | h::t -> if x = h then true
    else
    member x t;;

member 4 [1;2;4;6];;
#ocamlc -g member.ml
#rlwrap ocamldebug a.out

break:  break @filename linenumber
run
goto 0
print variables or p variables