Concept of Programming Languages (CS320)
Lecture 3

By Zhiqiang Ren
aren@cs.bu.edu
Content

- ATS Syntax Rephrase
- Operation on List and Tree
- Application of High Order Function
- Application of Closure
ATS Syntax Rephrase

- Expression: simple, compound, control flow expression
- Simple expression: literal, function call, object construction
- Compound expression:
  
  ```
  begin exp1; exp2; ...... ; expn end  // expn must be of type void
  (exp1; exp2; ...... ; expn)
  ```
- Control flow expression
  ```
  let ...... in ...... end
  case ... of
  if .... then ... else
  ```
- Pattern Match:
  ```
  val xxx = exp
  val () = !state := x
  ```
Operation on List and Tree

- List: operate from left to right
  - push_front is easy, push_back is difficult
  - append is difficult

- Tree: non-tail recursive is easy, tail recursive is difficult
Application of High Order Function

- Iterate
- Map
- Filter
- Comparison
Application of Closure

- Curry function: function with only one parameter
- Obj style
Mid-term sample

• // [listo_remove_dup] removes all the duplicates in a given list. E.g, // the list [1,7,2,1,3,5,3] becomes [1,7,2,3,5] after all duplications // are removed.

• //

• fun{a:t@ype} listo_remove_dup (xs: listo a, eq: (a, a) -> bool): listo a