Brady Dean

10/29/18

Lambda Calculus is made of expressions that are defined recursively:

$$< expression > := < name > | < function > | < application >$$
$$< funciton > := \lambda < name > . < expression >$$
$$< application > := < expression > < expression >$$

- The only two keywords are lambda and dot.
- Function application associates to the left.
- Identifiers that do not appear in the head are free variables.
    - $(\lambda x. xy)$ – y is a free variable
    - An identifier is free if it is unbound from an expression.
- All identifiers are local to their function.
    - $(\lambda x. x)(\lambda x. xy)$ – These are two distinct x's.
- If substituting E brings an unbound into a bound expression, the variable is renamed.