CS 162 Programming languages

Introduction to λ^+

Yu Feng

Winter 2021

Overview

- A λ^+ program is simply an expression
- Executing a program = evaluating an expression
- Constant expression: 8
- Arithmetic expression: (8 + 9) * 2

Let bindings

- Let bindings in λ^+ allow us to name and reuse expressions
- An expression of the form let x = e_1 in e_2:
  - bind the value of e_1 to identifier x and evaluates e_2 under this binding
    - e_1 is called the initializer
    - e_2 is referred to as the **body** of the let expression
- let x = 3+5 in x-2 evaluates to 6
- let x = 3+5 in x+y yields a run-time error due to undefined variable y

Let bindings

- Let bindings in λ^+ can be arbitrarily nested

```
let x = 3+5 in
let y = 2*x  in          ⟶        24
  y+x
```

```
let x = 2 in
let x = 3  in            ⟶        3
   x
```

Function definition

- Function definition:
  - **fun** f **with** x_1, ... x_n = e **in** e'
  - f is the name of the function being defined, x_1, ...x_n are the arguments of function f, and e is the body of function f

fun plus with x,y = x+y in
plus 2 3 ⟶ 5

fun fact with n =
if n=0 then1 else n*(fact (n-1)) in fact 3 ⟶ 6

List operations

- The empty list constant Nil
- The cons cell $e_1@e_2$, where e_1 is the head of the list and e_2 is the tail of the list
- The expression !e yields the head of the list. !(2@3@ Nil) evaluates to 2
- The expression #e yields the tail of the list. #(2@3@Nil) evaluates to 3@Nil

Lambda expressions