hw 6.

1. letrec

- problem: let fact = λn. if n = 0 then 1 else n * fact (n-1) in fact 5.

recall : let expression

$$\frac{e1 \to e1'}{\text{let } x=e1 \text{ in } e2 \to \text{let } x=e1' \text{ in } e2}\ [\text{e-let}]$$

⇒ evaluate RHS of binding

$$\frac{}{\text{let } x=v \text{ in } e2 \to e2\ [v/x]}\ [\text{e-letv}]$$

⇒ until RHS is a value, then substitute

$$\frac{G \vdash e1 : t1 \quad G, x:t1 \vdash e2 : t2}{G \vdash \text{let } x=e1 \text{ in } e2 : t2}\ [\text{t-let}]$$

⇒ type.

⇒ evaluating : fact 5

- add binding : (fact , e) to environment.
  - first evaluate e before adding the binding (should be value)
  - ⇒ evaluate e_1 found fact , but fact binding is not added.

⇒ solution : letrec

letrec fact = λn. if n = 0 then 1 else n * fact (n-1) in fact 5.

⚠ really exists : scheme, ocaml

letrec $f = \lambda x.e_1$ in $e \to e[(\lambda x.e_1)[\text{letrec } f=\lambda x.e_1 \text{ in } f / f] / f]$

- letrec $f = \lambda x.e_1$ in f
  - $\to f\,[(\lambda x.e_1)[\text{letrec } f=\lambda x.e_1 \text{ in } f / f] / f]$
  - $\to (\lambda x.e_1)[\text{letrec } f=\lambda x.e_1 \text{ in } f / f]$ ⇒ recursive !

$(\text{letrec } f = \pi x.e_1 \text{ in } f)\ v$

$\to (\pi x.e_1)[\text{letrec } f = \pi x.e_1 \text{ in } f/f]\ v$

$\overset{f \neq x}{=} (\pi x.\ e_1[\text{letrec } f = \pi x.e_1 \text{ in } f/f])\ v$

$\to e_1[\text{letrec } f = \pi x.e_1 \text{ in } f/f][v/x]$

$\overset{f=x}{=} \pi x.e_1$ (no free x)

$\Downarrow$

letrec $\equiv$ let