RECURSION

```
let rec concat a b = match a with
    | [] -> b
    | x::xs -> x::concat xs b
```

- to handle recursion, introduce type variables for the function

concat : $t_1 \rightarrow t_2 \rightarrow t_3$

- use these types to conclude the type of the body.

- pattern matching first case
  $[t_4] \rightarrow t_5 \rightarrow t_5$
  unify $[t_4]$ with t_1, t_5 with t_2,
  t_5 with t_3
  $t_1 = [t_4]$, $t_2 = t_3 = t_5$

- pattern matching second case
  $[t_6] \rightarrow t_7 \rightarrow [t_6]$

  unify $[t_6]$ with t_1, t_7 with t_2,
  $[t_6]$ with t_3

$\Rightarrow t_1 = [t_6] = [t_4]$ $\quad t_7 = t_2 = t_5$
$\Rightarrow t_6 = t_4$ $\quad [t_6] = t_3 = t_5$
$\Rightarrow t_5 = [t_4]$