Content

- Assignment V.S. Binding
- What does $delay$ do?
Assignment V.S. Bind (1)

```c
struct Foo
{
    int m;
    int n;
};

int main (int argc, char *argv[])
{
    int a = 3;
    a = 4;
    struct Foo s = {1, 2};
    int *p = &a;

    return 0;
}
```

Variable a is assigned the value of 3

Variable p is assigned the value of 359a3900

Variable s is assigned the value of struct {1, 2}

Variable is container.
implement main () = let
  val a = 3
  val b = a
  val b = 4
  val flat_t = @ (a, b)
  val box_t = '(a, b)
  val ra = ref_make_elt<int>(3)
  val rb = ra
  val () = printf("!ra is %d\n", @(!ra))
  val () = printf("!rb is %d\n", @(!rb))
  val () = print_newline ()
  val () = !rb := 4
  val () = printf("!ra is %d\n", @(!ra))
  val () = printf("!rb is %d\n", @(!rb))
in end

Name a is bound to the value of 3
Name b is bound to a then the value of 4
Name flat_t is bound to the value of (3, 4)
Name box_t is bound to the value of (3, 4)
Name rb is bound to the value of 45456700
reference

- datatype ltree =
  - | ltree_cons of (int, ref int, ref ltree, ref ltree)
  - | ltree_nil of ()

- val a: ref (ltree) = ref_make_elt<ltree> (3, ref<int> (4), rf_tl, rf_tr)
What does $\text{delay}$ do?

- delay the evaluation of expression
- modify the type of the return value (from stream\_con to stream)
Continuation

- abst@ype ans
- Assume ans = int

fun kf91 (x: int, k: int -<clorefi> int): int =
    if x > 100 then
        k (x - 10)
    else
        kf91 (x + 11, F (* lam x => kf91(x, k) *))
// end of [kf91]

\[
f'91(x) = f'91(f'91(x + 11))
\]

\[
k'f91(x, k) = k(f'91(x)) = k(f'91(f'91(x + 11)))
\]

\[
k'f91(x + 11, F') = F(f'91(x + 11))
\]

\[
F(x) = k(f'91(x)) = kf'91(x, k)
\]
terminating

- (*)
- Please construct a terminating function of the following type. Note that no use of exception is allowed.
- *)

- extern fun{a,b,c:type}
- S (f: (a, b) -> c, g: a -> b, x: a): c
- // end of [fun]

- implement {a,b,c} S (f, g) = S (f, g)