

Worksheet 10B

Quiz 1: Substitution

Suppose we have

```
sub0 := [a := int, b := bool, c := (-> (int int) int)]
ty0  := (-> (a z) b)
```

What is the result of *applying* sub0 to ty0?

Quiz 2: Substitution

Suppose we have

```
sub0 := [a := int, b := bool, c := (-> (int int) int)]
ty1  := (forall (a) (-> (a) a))
```

What is the result of *applying* sub0 to ty1?

Quiz 3: Free Variables of a Type

Fill in the implementation of free_vars which computes the set of variables that appear inside a (poly) type.

```
/* Free Variables of a Type */
fn free_vars(ty: &Ty) -> HashSet<TyVar> {
  match ty {
    int | bool =>

    a =>

    (-> (t0...) t) =>

    (forall (a0...an) t) =>

  }
}

/* Free Variables of a Env */
fn free_vars_env(env: &Env) -> HashSet<TyVar> {
  let mut res = HashSet::new();

  res
}
```

Quiz 4: Inference

In each case below, fill in the *inferred* type

env	[incr := (-> (int) int)]
subst	[]
expr	(incr 99)
infer(...)	

env	[id := (forall (a) (-> (a) a))]
subst	[]
expr	(id 10)
infer(...)	

env	[incr := (-> (int) int)]
subst	[]
expr	(fn (x) (incr x))
infer(...)	

env	[incr := (-> (int) int)]
subst	[]
expr	(fn (y) y)
infer(...)	

env	[incr := (-> (int) int)]
subst	[]
expr	(let (foo (fn (x) (incr x))) (foo 5))
infer(...)	

env	[incr := (-> (int) int)]
subst	[]
expr	(let (id (fn (y) y)) (id 9))
infer(...)	

Quiz 5: Your turn!

What is something you found confusing in today's lecture (or earlier)?