

CS 3721: Programming Languages Lab

Lab #08: Type inferencing and translating Scheme to ML

The following examples shows the result of type inference for a given Scheme function and the equivalent ML code.

```
(define apply2 (lambda (f x) (f (f x))))
```

```
(*  
  f: 'a -> 'a  
  x: 'a  
)*  
fun apply2(f,x) = f(f(x));
```

Statically determine a unique type for each of the variables in the following Scheme code. Based on the static type of each variable, translate the Scheme code to ML.

1. (define Repeat (lambda (f n x)
 (if (> n 1) (f (Repeat f (- n 1) x)) (f x))))
2. (define Map (lambda (f x)
 (cond ((null? x) '())
 (else (cons (f (car x)) (Map f (cdr x)))))))
3. (define reduce (lambda (f1 f2 x)
 (cond ((null? x) '())
 (else (f2 (f1 (car x)) (reduce f1 f2 (cdr x)))))))
4. (define main
 (let ((apply2 (lambda (f x) (f (f x))))
 (time2 (lambda (x) (* 2 x))))
 (apply2 time2 2)))