1 Lambda lifting

```
let g = 22;
fun add a = a + g;
```

Listing 1: Simple add in ML

$$\begin{aligned} add' &= \lambda g.\lambda a.a + g \\ add &= add' g \end{aligned} \tag{1}$$

```
let g = 22;
fun r a = r (a + g);
```

Listing 2: Recursive function in ML

$$\begin{aligned} r' &= \lambda r.\lambda g.\lambda a.r(g)(a + g) \\ r &= (r' r')(g) \end{aligned} \tag{2}$$

```
let g = 22;
fun f a = y (a + g);
fun y a = f (a + g);
```

Listing 3: Mutually recursive function in ML

$$\begin{gathered} f' = \lambda f.\lambda y.\lambda g.\lambda a.y\ y\ f\ g\ (a + g) \\ y' = \lambda y.\lambda f.\lambda g.\lambda a.f\ f\ y\ g\ (a + g) \\ f = f'\ (f')\ (y') \\ y = y'\ (y')\ (f') \\ \text{or} \\ f' = \lambda y.\lambda g.\lambda a.y\ g\ (a + g) \\ y' = \lambda f.\lambda g.\lambda a.f\ g\ (a + g) \\ f = f'\ (y'\ f') \\ y = y'\ (f'\ y') \end{gathered} \tag{3}$$

```
fun m =
    fun f a = y a;
    f
fun y a = m a;
```

Listing 4: Mutually recursive closure in ML

$$\begin{aligned} m' &= \lambda y.\lambda a.y\ a \\ y' &= \lambda m.\lambda a.m\ a \\ y &= y'\ m' \\ m &= m'\ y' \end{aligned} \tag{4}$$

```
let q = 11;
fun f a =
    let x = 22;
    fun g b = a + x + b + q;
    g;
```

Listing 5: Closures in ML

$$\begin{aligned} g' &= \lambda q\lambda a.\lambda x.\lambda b.a + b + x + q \\ f' &= \lambda q.\lambda a.g'\ q\ a\ 22 \\ f &= f'\ 11 \end{aligned} \tag{5}$$