Haskell Program

```
x :: (a -> b -> c) -> [a] -> [b] -> [c]
x = zipWith
```

A. *Original Constraints*

```
x(V, _) ←  V = fun(V_1 , V_2 ),
           V_1  = fun(V_a , V_3 ),
           V_3  = fun(V_b , V_c ),
           V_2  = fun(V_4 , V_5 ),
           V_4  = list(V_a ),
           V_5  = fun(V_6 , V_7 ),
           V_6  = list(V_b ),
           V_7  = list(V_c ),
           zipWith(V, _).
```

B. After Constraint-Combining

```
x(V, _) ←  V = fun(
             fun(
               V_a ,
               fun(V_b , V_c )),
             fun(
               list(V_a ),
               fun(list(V_b ), list(V_c )))),
           zipWith(V, _).
```