



### Aufgabe 14.3: Übersetzung und Rückübersetzung (3+7)

```
(a) fun compile' (Con i)      = [con i]
    | compile' (Add(e1,e2)) = compile' e2 @ compile' e1 @ [add]
    | compile' (Sub(e1,e2)) = compile' e2 @ compile' e1 @ [sub]
    | compile' (Mul(e1,e2)) = compile' e2 @ compile' e1 @ [mul]

    fun compile e = compile' e @ [halt]

(b) fun decompile' (con n::is, es)      = decompile'(is, Con n::es)
    | decompile' (add::is, e::e'::es) = decompile'(is, Add(e,e')::es)
    | decompile' (sub::is, e::e'::es) = decompile'(is, Sub(e,e')::es)
    | decompile' (mul::is, e::e'::es) = decompile'(is, Mul(e,e')::es)
    | decompile' ([halt], [e])        = e
    | decompile' _                     = raise Error "cannot decompile"

    fun decompile code = decompile'(code,nil)
```

### Aufgabe 14.4: Fibonacci-Prozedur in V (5+5)

```
(a) [proc(1,17),
     con 1, getF ~1, leq, cbranch 3,
     getF ~1, return,
     con 1, getF ~1, sub, call 0,
     con 2, getF ~1, sub, call 0,
     add,
     return]

(b) [proc(3,15),
     con 0, getF ~1, leq, cbranch 3,
     getF ~2, return,
     getF ~3, getF ~2, add, getF ~3, con 1, getF ~1, sub, callR 0,
     proc(1,5),
     con 1, con 0, getF ~1, callR 0]
```

### Aufgabe 14.5: Verschränkte Rekursion in V (10)

```
[proc(1,9),
 getF ~1, cbranch 5,
 con 1, getF ~1, sub, callR 9,
 con 1,
 return,
 proc(1,9),
 getF ~1, cbranch 5,
 con 1, getF ~1, sub, callR 0,
 con 0,
 return]
```