

# Compiler Construction

~ Static Links ~

# What are Static link

## Goal

How to handle escaping variables and recursion?

## Static Links

A mechanism:

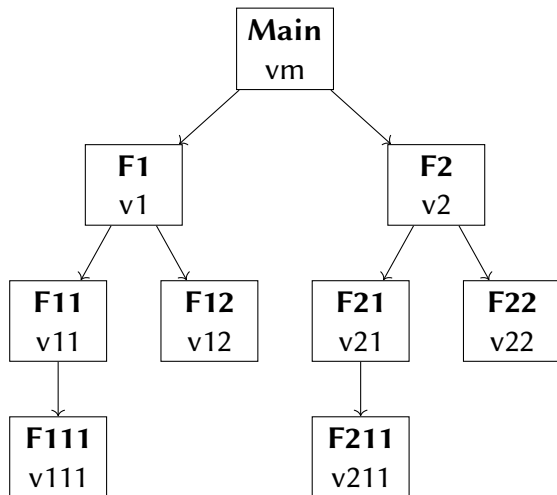
- computed a compile time
- that produces code
- that will be used at runtime

## escapes-n-recursion

```
let function trace(fn: string, val: int) =  
    (print(fn); print("("); print_int(val); print(") "))  
  
    function one(input : int) =  
        let function two() =  
            (trace("two", input); one(input - 1))  
        in  
            if input > 0 then  
                (two(); trace("one", input))  
            end  
        in  
            one(3); print("\n")  
    end
```

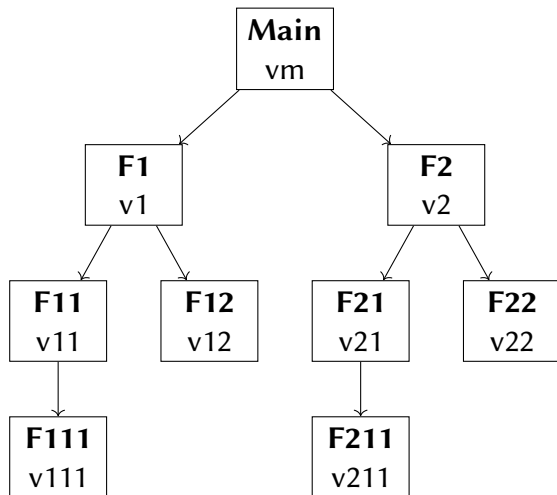
```
two(3) two(2) two(1) one(1) one(2) one(3)
```

# Deep Static Function Hierarchies



# Deep Static Function Hierarchies

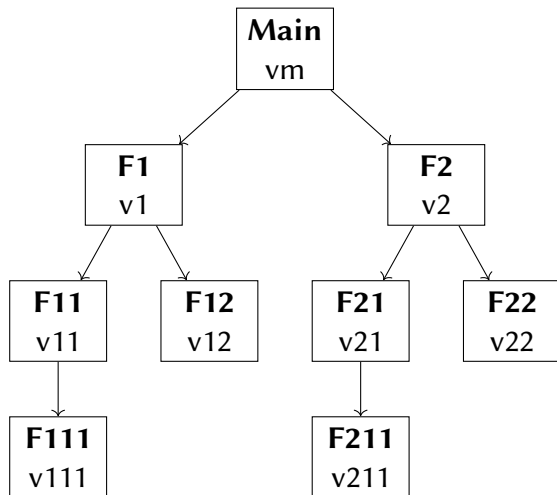
What if:



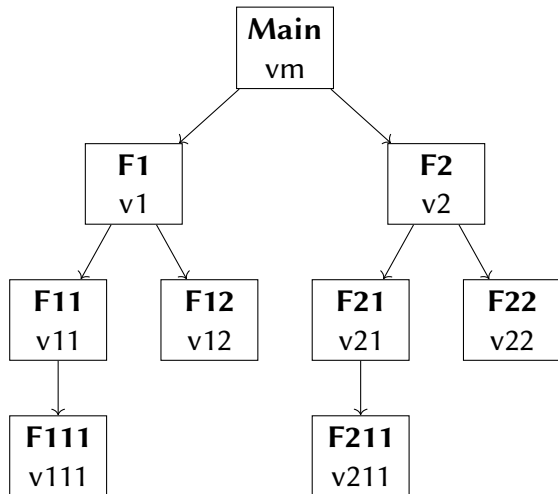
# Deep Static Function Hierarchies

What if:

- 1 **Main** uses **vm**



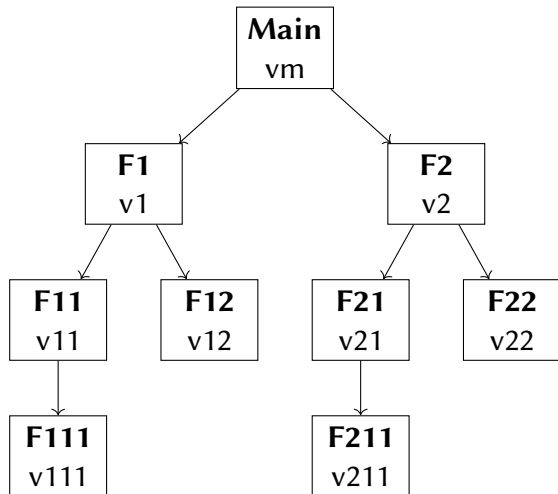
# Deep Static Function Hierarchies



What if:

- 1 **Main** uses **vm**
- 2 **Main** calls **F1**

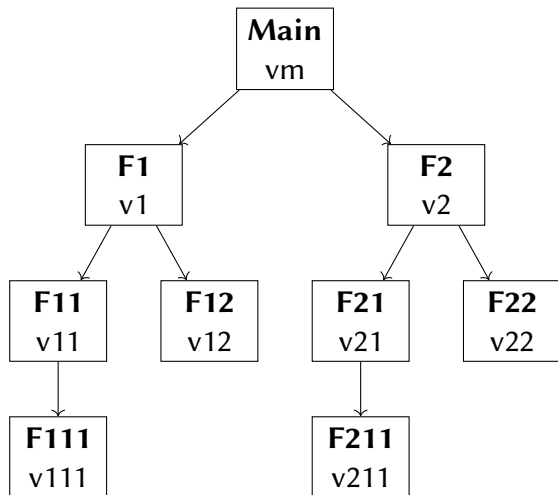
# Deep Static Function Hierarchies



What if:

- 1 **Main** uses **vm**
- 2 **Main** calls **F1**
- 3 **F1** uses **v1**

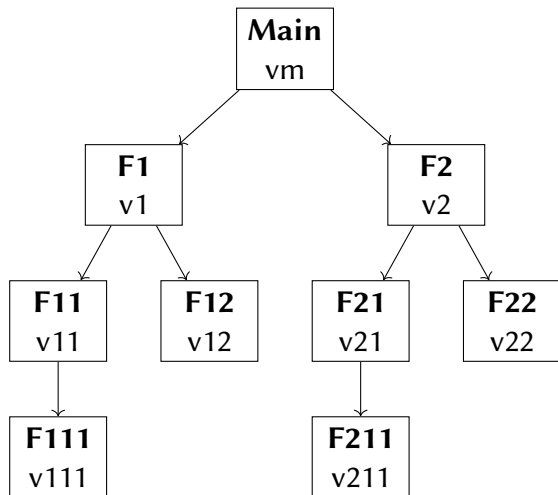
# Deep Static Function Hierarchies



What if:

- 1 **Main** uses **vm**
- 2 **Main** calls **F1**
- 3 **F1** uses **v1**
- 4 **F1** uses **vm**, non local

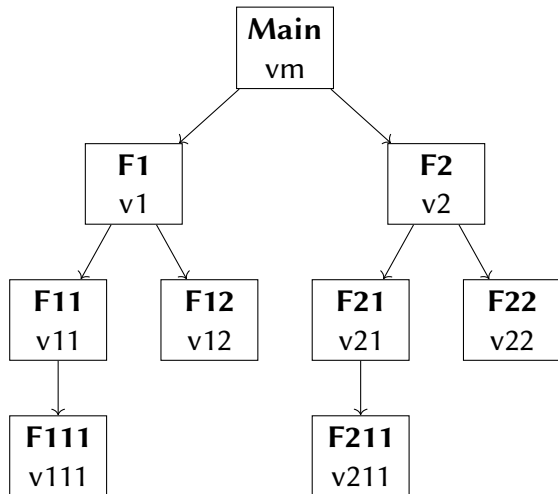
# Deep Static Function Hierarchies



What if:

- 1 **Main** uses **vm**
- 2 **Main** calls **F1**
- 3 **F1** uses **v1**
- 4 **F1** uses **vm**, non local
- 5 **F1** calls **F11**

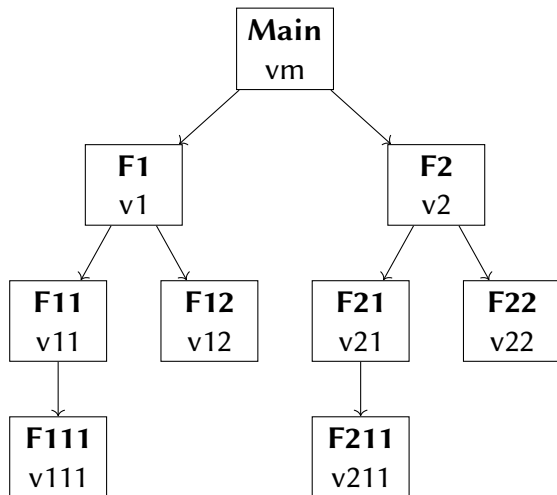
# Deep Static Function Hierarchies



What if:

- ① **Main** uses **vm**
- ② **Main** calls **F1**
- ③ **F1** uses **v1**
- ④ **F1** uses **vm**, non local
- ⑤ **F1** calls **F11**
- ⑥ **F11** uses **v11**

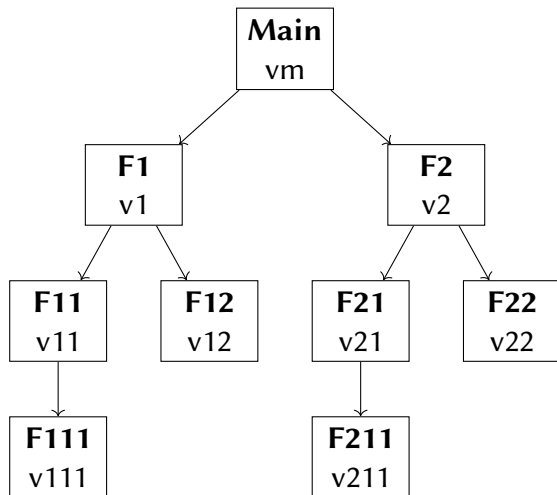
# Deep Static Function Hierarchies



What if:

- ① **Main** uses **vm**
- ② **Main** calls **F1**
- ③ **F1** uses **v1**
- ④ **F1** uses **vm**, non local
- ⑤ **F1** calls **F11**
- ⑥ **F11** uses **v11**
- ⑦ **F11** uses **v1**

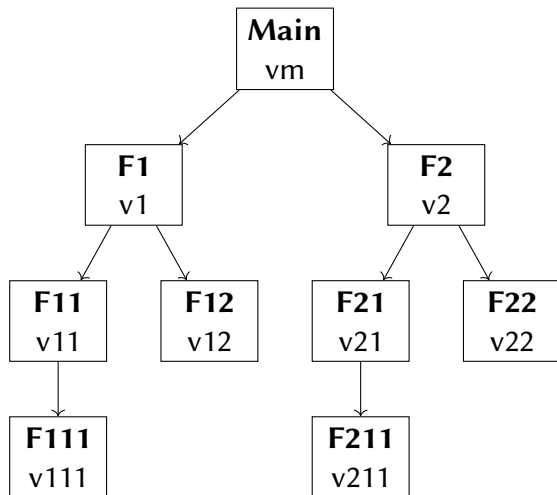
# Deep Static Function Hierarchies



What if:

- ① **Main** uses **vm**
- ② **Main** calls **F1**
- ③ **F1** uses **v1**
- ④ **F1** uses **vm**, non local
- ⑤ **F1** calls **F11**
- ⑥ **F11** uses **v11**
- ⑦ **F11** uses **v1**
- ⑧ **F11** uses **vm**

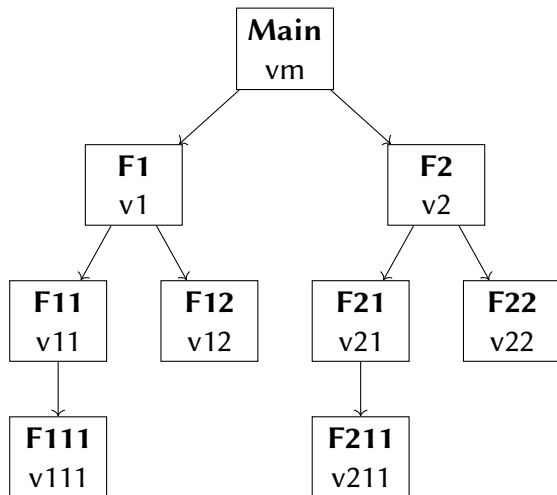
# Deep Static Function Hierarchies



What if:

- ① **Main** uses **vm**
- ② **Main** calls **F1**
- ③ **F1** uses **v1**
- ④ **F1** uses **vm**, non local
- ⑤ **F1** calls **F11**
- ⑥ **F11** uses **v11**
- ⑦ **F11** uses **v1**
- ⑧ **F11** uses **vm**
- ⑨ **F11** calls **F12**

# Deep Static Function Hierarchies



What if:

- 1 **Main** uses **vm**
- 2 **Main** calls **F1**
- 3 **F1** uses **v1**
- 4 **F1** uses **vm**, non local
- 5 **F1** calls **F11**
- 6 **F11** uses **v11**
- 7 **F11** uses **v1**
- 8 **F11** uses **vm**
- 9 **F11** calls **F12**
- 10 **F12** calls **F1**

# Deep Static Function Hierarchies

The caller must provide the callee with its static link.


Caller	Callee	Static Link
Main	F1	$fp_{Main} = fp$
F1	F11	$fp_{F1} = fp$
F11	F12	$fp_{F1} = sl_{F11} = *fp_{F11} = *fp$
F12	F2	$fp_{Main} = sl_{F1} = *sl_{F12} = **fp_{F12} = **fp$
F2	F22	$fp_{F2} = fp$
F22	F11	$fp_{F1} = ???$

Assuming that the static link is stored at  $fp$ .

# Summary



Static Link



Escaping  
Variables