3 Closure representations

Closure

A closure is:
- Code
- Environment

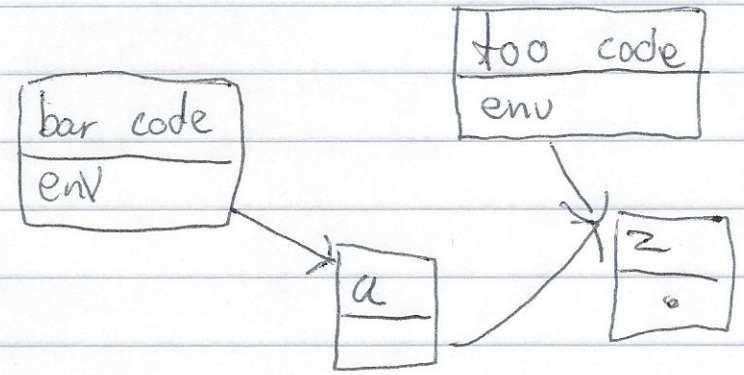
Why is closure needed?

Variables can escape: outlive their stack frame.
~~Stack must~~ Environment must be decoupled from the stack.
Example:

```
function foo(a)
    return function()
        return a
    end
end
```

Deep closure

Linked list of environments.

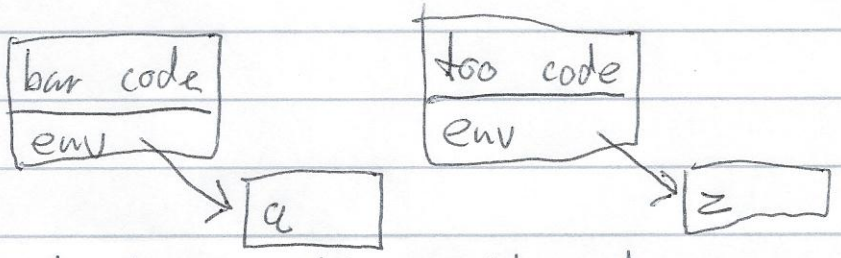


+ Simple
+ Fast to create

÷ May create memory leaks
÷ Outer variables require search

Flat closure

Contains copy of each free variable

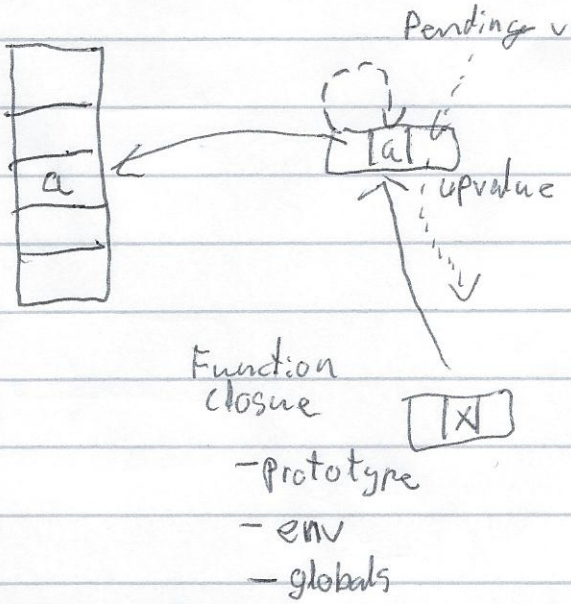


∴ assignment to variable might not be visible elsewhere
- Typically handled by boxing

Daimi scheme has both, it is up to the compiler to choose.

LUA

Uses flat closures with indirection.



+ Non users will not pay for the level of indirection

When a closure is created it searches for outer ~~values~~ variables in pending vars. If not found a new upvalue is created.

Upvalue is closed when stack is being destroyed.