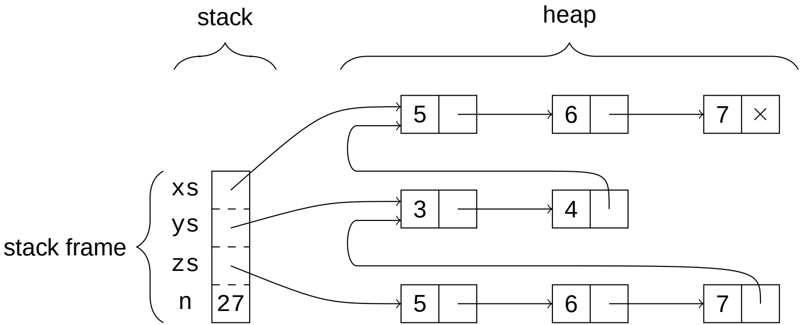
Memory management: stack and heap

- Primitive values are allocated on the stack
- Composite values are allocated on the heap

```
let xs = [5;6;7];;
let ys = 3::4::xs;;
let zs = xs @ ys;;
let n = 27;;
```



Observations

No unnecessary copying is done:

1. The linked lists for *ys* is not copied when building a linked list for *y* :: *ys*.
2. Fresh cons cells are made for the elements of *xs* only when building a linked list for *xs* @ *ys*.

since a list is a functional (immutable) data structure

The running time of @ is linear in the length of its first argument.