

0-1) CEK

$$\langle X, \Sigma, K \rangle \mapsto \langle V, \Sigma, K \rangle \quad \text{where } [X \mapsto V] \in \Sigma$$

$$\langle M, N, \Sigma, K \rangle \mapsto \langle M, \Sigma, \text{fn}(N, \Sigma, K) \rangle$$

$$\langle V, \Sigma, \text{fn}(N, \Sigma', K) \rangle \mapsto \langle N, \Sigma', \text{an}(V, K) \rangle$$

$$\langle \lambda X.M, \Sigma, K \rangle \mapsto \langle \text{do}(\lambda X.M, \Sigma), \Sigma, K \rangle$$

$$\langle V, \Sigma, \text{an}(\text{do}(\lambda X.M, \Sigma'), K) \rangle \mapsto \langle M, \Sigma' [X \mapsto V], K \rangle$$

Memory Management — obvious vs hard

malloc at beginning, free at end — easy

trees and lists and stuff — hard

easy to track life → named variable

move semantics + initialize in C++

↳ when copied vs "same thing"

unnamed memory → hard

order of destruction hard (recur)

variables are names for values

↳ x, y, z ↳ 5, 6, "Say"

locations store values (and are values in C)

x = x8088 → 5

y = x8088

vars locs vals

z = x8080 → x0890

x8090 → 7

easy — one var, one loc, one val } stack management

— done w/ var = done w/ loc

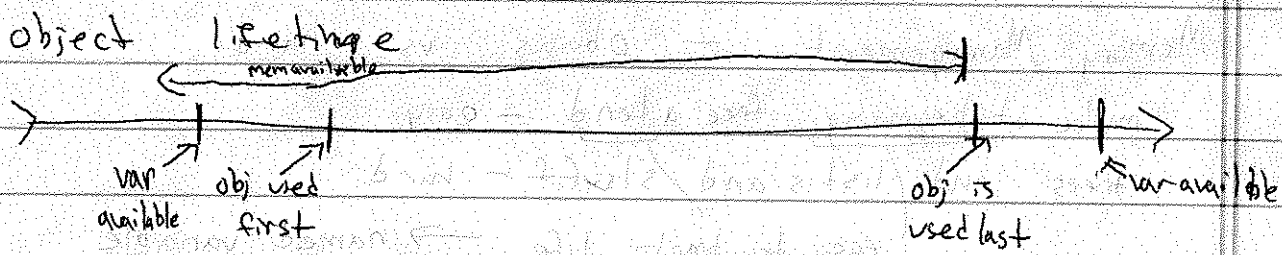
hard — more vars than locs (aliasing) (x = 10, y = 10)

fewer vars than locs (heap shape analysis) (x = 10, 11)

10-2/ Hand needs... something else > memory management

correctness

Soundness - if the model does it, the theory does - all actions justifiable
 completeness - theory -> model - you take all just actions



```

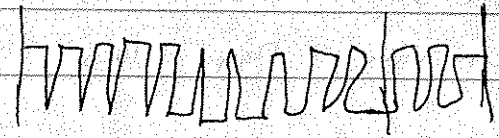
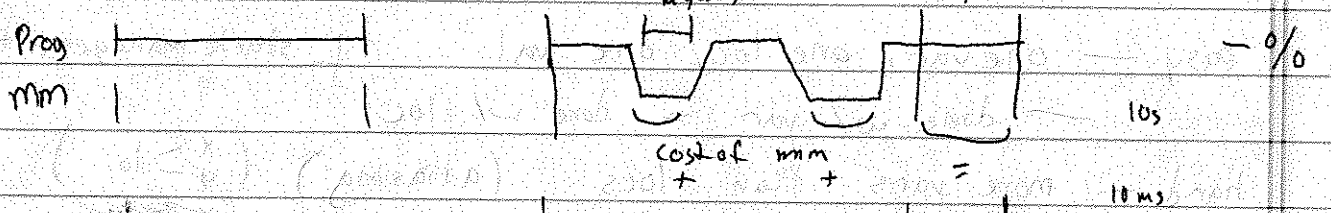
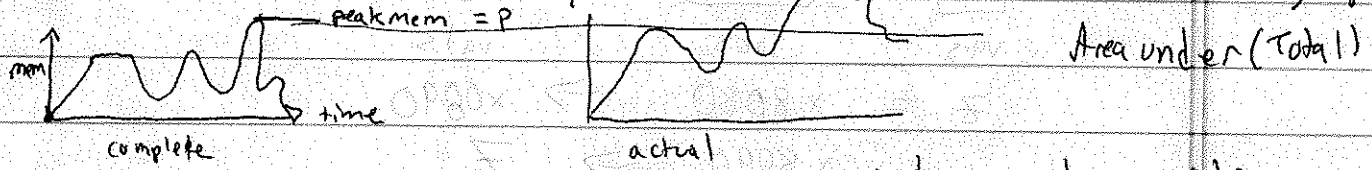
int x = 8;
... 200 times ...
y = x + 3;
... 300 times ...
ret y;
}

if ( f(y) ) {
    x = x + 1;
}

f = does this TM halt?
    
```

MM completeness -> is possible!
 incomplete means need X mem but you use X+4 mem
 unsound means => program crashes

Optimize perf unrestrained by soundness



10-3

Manual (ie call malloc/free)

- Sound? If you do it right, i.e NO
 But possible to do right.

- Costs?

↳ Programmer increases, you put in calls (maybe 150-200)

↳ memory → calling m/f take stack space

- tend to make copies vs aliases 150-200%

↳ time → $\lg N$ (where N is memory) - malloc



$2 \times \lg N$

- free



