```
(it's used exactly once)
     Rust & Clean can enforce linearity
     CESK - machine
                                            K= m+
                        L7 Store
     Lti Code
                                             1 < fn, v, k>
       L7 environment
                        > Continuation
                                            ) <ar, c, E, K>
                                              1 cop , ... >
                                       new > 1 < set, o, k>
           < (MN), E, S, K7
            HO < M, E, S, < ar, N, E, K>>
          c (call/cc M), E, S, K>
aside on
           H> < (M (cont, K>), E, S, K> (let X = 99 in
Total-(ESK
          < V, E, S, < ar, N, E', K>> ( (begin (set! X5)
            \mapsto \langle N, \epsilon', S, \langle fn, V, k \rangle \rangle add 1)
          < 1X,1M, E, S, K> >> < < (6,1X,1M, E>, E, S, K> X))
          < V, E, S, < fn, < clo, \( \lambda \text{X.M}, \( \ext{E'} \rangle , \text{K} \rangle \) = 7 6
           HO < M, E'[XHO], S[OHV], K)
                 where o & dom(s)
          < (set! x m), E, S, k>
            H> < M, E, S, < set, E(X), K>>
          < V, E, S, < set, o, K>>
          H> < void, E, S[OH>V], K>
           interpret: env expr -> value
          store-interpret : env store expr -> (value, store)
            [(add lhs rhs)]
(lhsv, lhs &) != sii, (E, E, lhs)
               (rhsv, rhs { != sii (E, hs E, rhs)
               (c+ lhsv rhsv), Ths E)
```

The Store ( & ) is a linear variable

7-1/

```
19-2/ Rules for C pointers!
       free nust be called exactly once
     How to Enforce
        Every function either
            (i) weeks frees the pointers it is given
            (2) returns them to caller
       (1) To gave Hem to such a function Los actually call free
       (2) = literally return

read-only (not copying)
     3 07
       p = malloc (i)
        g (p) casel: g only reads pieces of p
                    we still "own" p
                 case Z: of frees p
                      we can't mention it again
  Goal: (+26), [5, +74]>
             H→ < (+ Z 6) , Ø >
    (M, \Sigma[\sigma, \mapsto V,], [\sigma_n \mapsto V_n] > [gc-rule]
        H> < M, E>
           J, ... Jn € LS ( < M, € > )
  Live Set LS(<M, E>) = LS(M) U 2S(E)
         LS([o] HV]...[on HVn]) = U LS(Vi)
  LS(x) = \emptyset
LS(\sigma) = \xi \sigma 
LS(\lambda X, m) = LS(m)
  LS (MN) = LS(M) ULS(N) LS (X:=M) = LS (M)
```

| 9-3/   | < C, E, S, K> < C, E, S', K>                                      |
|--|---|
|  | Siti glubale glocals shock   S'= E(O, S(O) > lo E L ]             |
|  |   |
|  | Ø , S >   |
|  | ⊢Fige.  |
|  | (0, 4,5)  |
| since region in Alliance are in decident and Alliand Alliand Alliana regions, years produced and an alliana an  |   |
| 14 data Sawakon nggaran sayayo wandiga hijin kao wa ambou hi kan ang aligung wasawa ka dida saya ka  | $\rightarrow ge: \langle Set(\sigma), Set(\sigma), Store \rangle$ |
|  | gray black<br>live, pointing or live root set                     |
|  | live, pointing or live root set                                   |
|  | processed   |
| thread the first and the control of the first and the firs |   |
|  | LL(m) = LS(m)   |
|  | LL(E=[Xo H) Oo] [Xn H) On])                                       |
|  | = 200, 111, on 3  |
|  | $LL(m+) = \emptyset$ $LL(\langle fn, v, k \rangle) =$             |
|  | LL(V) U LL(K)   |
|  | LL (car, c, E, K>) =  |
|  | LL(C) ULL(E) LL(K)  |
|  | LL ( <set, k="" o,="">) =</set,>                                  |
|  | LL(k) v 203   |
|  | subtley wrong   |
|  | <6, B, S > <6/00/B ULL(S(00)),                                    |
|  | 00 6 G B U 8003, S>   |
|  | (GULL(S(OO))) \ (8000)  |
|  | Proofs:   |
|  | - constitent (deterministic)                                      |
|  |   |
|  | - completes/halts (doesn't get trapped m cycles)                  |
|  |   |
| , with whole detailed in SSS in the interview we work with a MSS in the SSS of the SS    |   |
|  |   |
|  |   |

