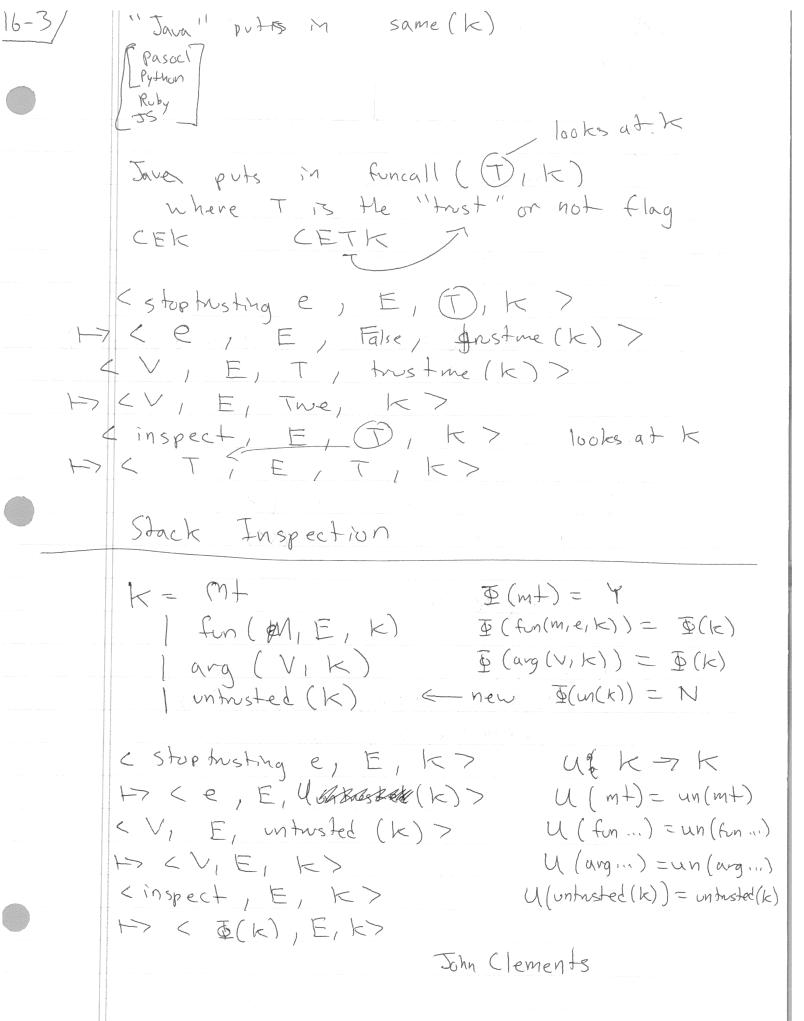
public static voil man (String [] args) { retern main (args); => Stack Overflow main (ary sigpend(args)) at Ctimain (tijava: 3) x (100 to 1033) CEK -> continuation Ly code Jenv Stack
Heap D= (Axixx) (Axixx) $\langle (\lambda f, (\lambda x, f(x \times))(\lambda x, f(x \times)))(\lambda y, y), \phi, m+ \rangle$ \rightarrow 3 (Ax, f(xx)) (Ax, f(xx)), [f \mapsto clo(Ay, y, d)]=E, m+> < (1x, f(xx))=wf I, fun((1x,f(xx)), E, m+)> < clo(wf, E,), E, fun (wf, E,, m+) > < < (lo(me, E,), E,, arg (clo (me, E,), mt) > $\leq \Omega$, ϕ , m+7with constant $\langle w, \phi, fun(w, \phi, mt) \rangle$ heap + $\langle clo(\omega, \phi), \phi \rangle$ stack 4 w, Ø, arg(clo(w, Ø), m+)> Space < x x, Ø[x+)clo(w, Ø)]=E, m+ 7/5 @<XX, Ø[XHO(W,Ø)], m+>

16-2 $\langle V, E, arg(clo(x,e,E'), k \rangle$ F) < e, E'[XI->V], K> CEK rule "A CEK funcall reduces stack space" Javas H> < e, E'[XH>V], same(K)> Rule $\langle V, E, same(k) \rangle$ New rule HY CV, E, K> $\langle w, \emptyset, arg(clo(w, \emptyset), m+) \rangle$ < clo(w,d), arg (clo(v,d), m+)> < x x, \$\(\pri \) \(\text{X} \) \(\text{Clo}(\omega, \pri) \) \] \(\text{Some} \) \(\text{M} \) \(\text{M < x / E, , fun(x, E, , same (m+) }> < X, E, , arg (clo(w,d), same (m+))> (A') < X X, E, , same (same (m+)) > Java ["not safe-for -space" 13 L" do not have proper fail-calls" CEK - ('tail-call optimizing" The stack is for evaluate function arguments NOT calling functions (A function is a parameterized 6070) 3 (x) 7 +mi return 1 + f(x)



cd sre &B make c d sre d∂ cg! make DD cd ... if [-2 sne] cd src DA (make; K = , . . . with-cud(d, e) a cud (d, k) read-cud() Generalize ... with-continuation -mark (K, v, e) I read-combingtion-marks (1) K = ... wem (N,V, K) $U(V_k, V_k, K)$ Ze U(VK,VV, WCM(VK,VVI,K)) = wem(VK, VV, K)