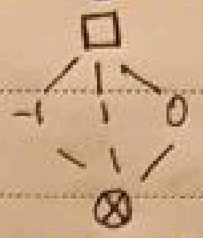
Subject:
Date

یک الگوریتم: برای شروع، $C(s, x, in) = \square$ state s، بقیه $C(n,x,in) = C(n,x,out) = \otimes$ همجا

مقداردهی اولیه که اینو میدیم برای حل کردن loopها

Orderings we can simply The presentation of The analysis by ordering The values. $\otimes < C < \square$ → stands for any possible run time value.

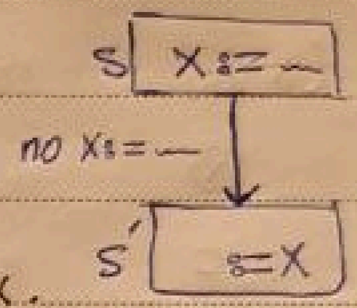


lub: least-upper-bound $lub(1,2) = \square$, $lub(\square, \otimes) = \square$, $lub(1, \otimes) = 1$

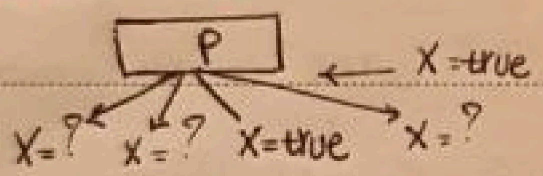
$C(s, x, in) = lub\{C(p, x, out) \mid p \text{ is a predecessor of } s\}$

Liveness A variable X is live at statement s if

- There exists a statement s' That uses x.
- There is a Path from s to s'.
- That Path has no interrening assignment to x.

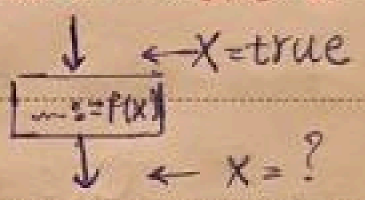


Liveness rule 1



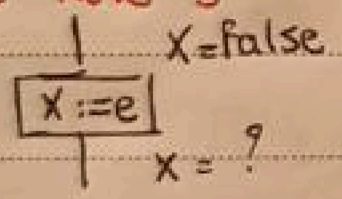
$L(p, x, out) = \vee\{L(s, x, in) \mid s \text{ a successor of } p\}$

liveness Rule 2



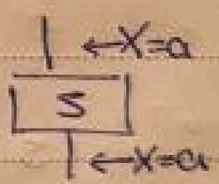
$L(s, x, in) = true$ if s refers To x on The rhs

Liveness Rule 3



$L(x:=e, x, in) = false$ if e does not refer To x

Liveness Rule 4



$L(s, x, in) = L(s, x, out)$ if s does not refer To x.

الگوریتم: همه رو false بذار بعد برو ببین کجا زنده شده از همون جا rule بزن

* liveness is backward analysis.

* if $w := rhs$ appears in a basic block, w does not appear any where else in Program

Then the statement $w := rhs$ is dead and can be eliminated.

$x := z+y$ $a := x$ $x := 2*a$	⇒	$b := z+y$ $a := b$ $x := 2*a$	⇒	$b := z+y$ $a := b$ $x := 2*b$	⇒	$b := z+y$ $x := 2*b$
	turn To single assignment form		copy Propagation		dead code elimination	

Global Data Flow analysis

Global constant Propagation → it can be performed at any point where ** holds

(**) On every path To the use of x, the last assignment to x is $x := k$

⊗ This statement never executes

c x = constant c

□ x is not a constant

* For each statement s, we compute information about the value of x immediatly before and after s.

Stands for 'constant' information

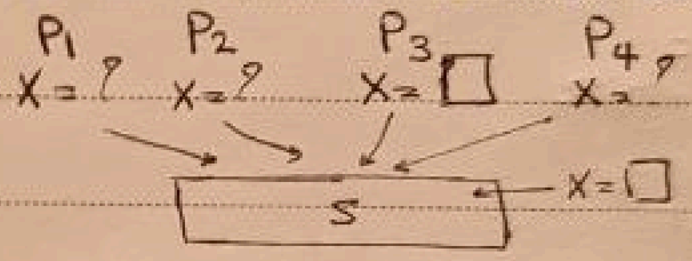
$C(s,x,in)$ = value of x before s is executed.

$C(s,x,OUT)$ = value of x after s is executed.

فرض کنید $P_1, \ldots, P_n$
پدرهای s باشند

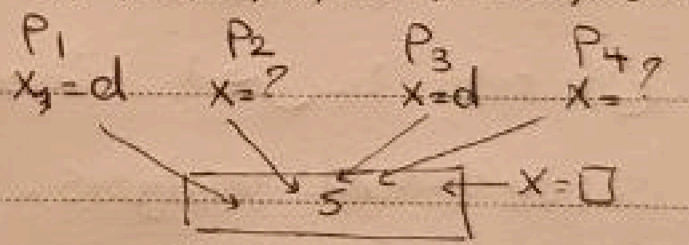
OUT → IN

Rule 1



if $\exists i\ (C(P_i,x,OUT)=\square)$, Then $C(s,x,in)=\square$

Rule 2



if $\exists_{i,j}\ (C(P_i,x,OUT)=c\ \&\ C(P_j,x,OUT)=d\ \&\ d<>c$

then $C(s,x,in)=\square$

Rule3 if $\forall_i\ (C(P_i,x,OUT)=c^{\text{or } \otimes})$

then $C(s,x,in)=c$

Rule4 if $\forall_i\ (C(P_i,x,OUT)=\otimes)$

then $C(s,x,in)=\otimes$

IN → OUT

Rule 5 if $C(s,x,in)=\otimes \Rightarrow C(s,x,OUT)=\otimes$

Rule 6 if c is constant $\Rightarrow C(x:=c,x,OUT)=c$

Rule 7 if $f(\ldots)$ is anything except a constant $C(x:=f(\ldots),x,OUT)=\square$

Rule 8 if $x<>y \Rightarrow C(y:=\ldots,x,OUT)=C(y:=\ldots,x,in)$