

Reaching Definitions \leftarrow can work without SSAreaches(l, x, l') — definition of temp x at line l reaches l'

$$\frac{\text{def}(l, x) \text{ succ}(l, l')}{\text{reaches}(l, x, l')}$$

$$\frac{\text{reaches}(l, x, l') \neg \text{def}(l', x) \text{ succ}(l', l'')}{\text{reaches}(l, x, l'')}$$

Complexity: $O(\# \text{lines}^2)$

Algorithm: propagate forward from definition

Optimisation

Optimise:

runtime	main thing]
mem use	for us	
code size		
energy use	want to improve average case	

Reg alloc optim

5. Coalesce non-interfering moves

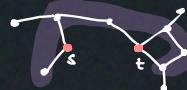
i.e. $t \leftarrow s$ if t and s do not interfere
so t and s can use same reg↳ get $t \leftarrow t$, useless move

↳ eliminate

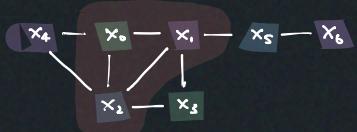
Greedy Coalescing

for each move $t \leftarrow s$:if $\{t, s\} \in E$:

skip

else if \exists colour $c \notin \text{col}(\text{N}(s) \cup \text{N}(t))$ that corresponds to a register:coalesce s and t into fresh temp w add w to graph with color w connect w to $\text{N}(s) \cup \text{N}(t)$ remove s and t replace s and t with w 

Ex.

 $N(x_4) \cup N(x_3)$  $x_0 \leftarrow x_4$ $x_4 \leftarrow x_3$

skip

can coalesce

 $3 \leq 13$, coalesce needs new color

* Move examination order affects result, so maybe check important parts first

* Coalescing preserves SSA