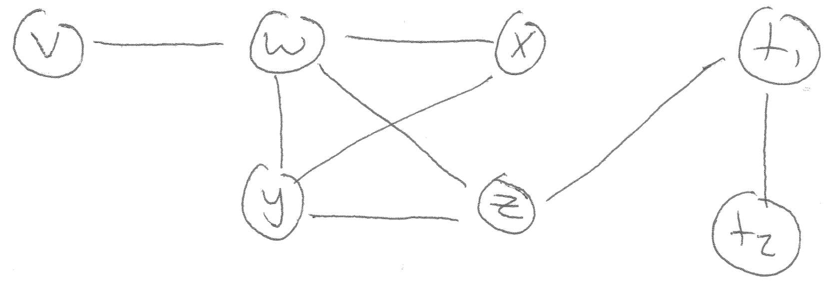
5-1

I interference



Assign each variable a register $(M : V \to R)$
s.t. no two variables adjacent in I
have the same mapping

$\forall (u, v) \in E \,.\, M(u) \neq M(v)$

feasible : A valid solution
optimal : Most preferred

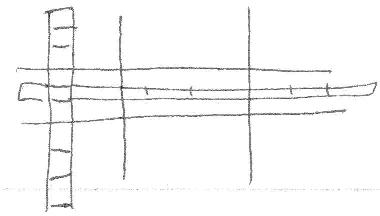
Max clique size is the minimum registers if > # of regs
$\Longrightarrow$ no feasible

$M' : V \to N$ feasible : some assignment
optimal : smallest (minimize maximum $M(v)$)

If $M'(v) = i$ and $i <$ # of regs, then use reg_i
o.w. use stack location (i - # of regs)

Graph Coloring

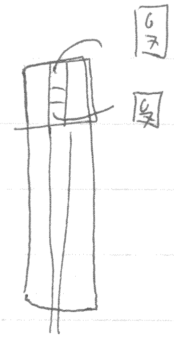
5-2



a number can only appear once per row, column, or box $\in [1, 9]$

1	2	3
4	5	8
	7	9

9-coloring



Saturation - algorithm (Pencil Marks)

Every node u has a set $sat(u)$

$= \{ c \in N \mid \exists v.\ v \in adj(u) \text{ and } color(v) = c \}$

Options for a node u are just $REGS - sat(u)$

$N - sat(u)$

1. Pick a v where $M'(v) =$ [illegible]
2. Choose the minimal option for v, $M'(v) := c$
3. Update sat of $adj(v)$
4. Repeat 1

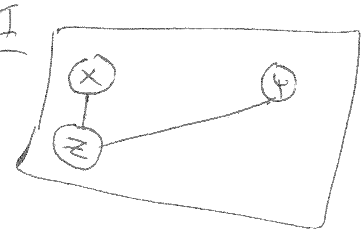
What order to visit nodes?

- connected (how many neighbors)
- more constraints / fewer options
- follow propagation
- useful variables (used-a-lot) [loops]

5-3

move - biasing

I



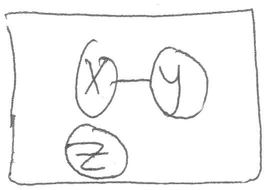
(movq (var x) (var y))

⇓

(movq (reg 0) (reg 1)) ⟹ (movq (reg 0) (reg 0))

Build a move bias graph, M

M



2: Choose an M-minimal option

X86 - registers

rax rcx rdx rsi rdi r8 r9 r10 r11] caller

rbx rsp rbp r12 r13 r14 r15] callee

caller-saves ⟵ before call, must save

callee-saves ⟵ fun <u>called</u> must save/restore