Computational Foundations for ML

10-607

Geoff Gordon

Notes and reminders

- Lab 2 due today
- HW 1 due Wednesday

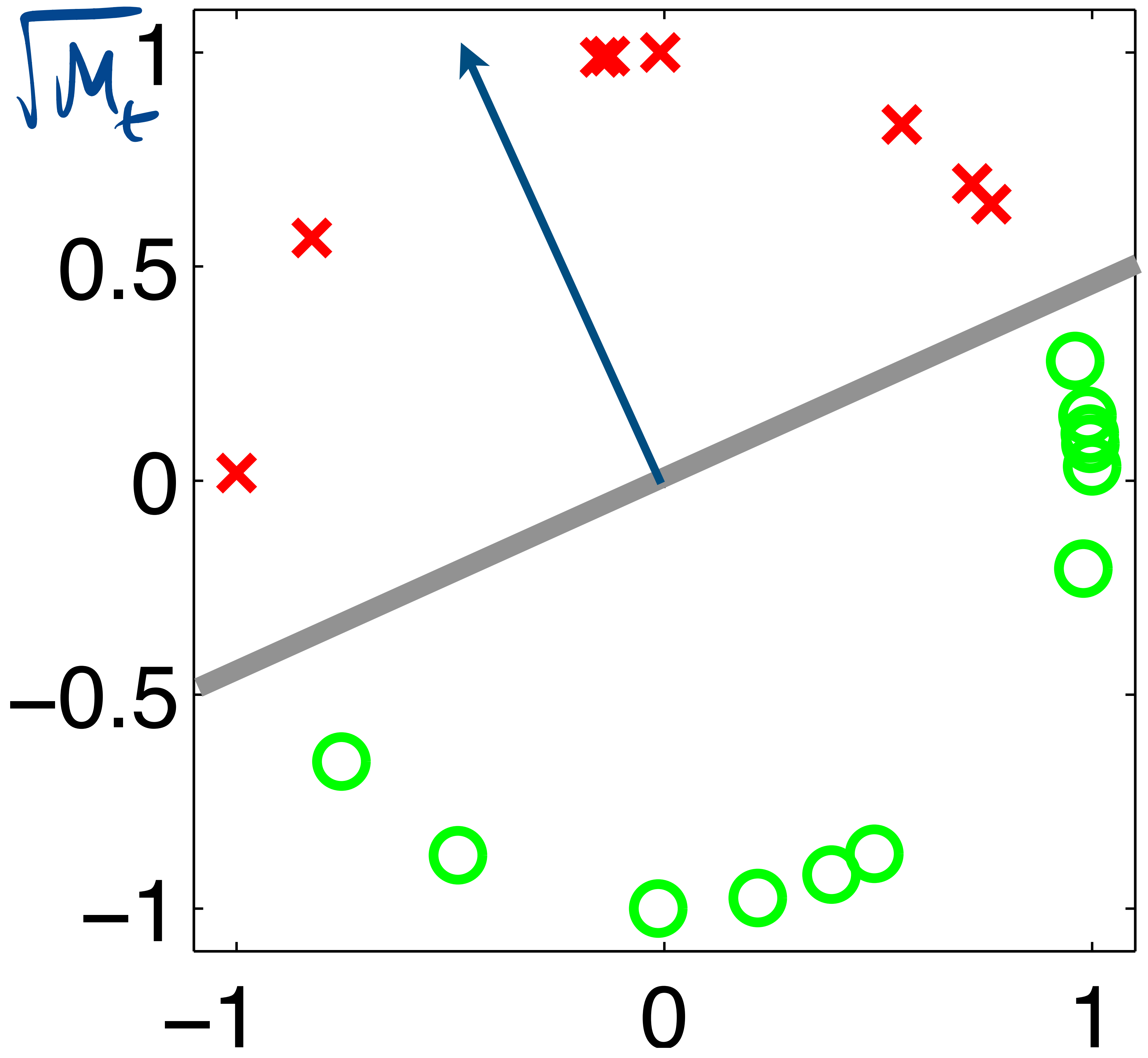
Perceptrons

I $\quad w_t \cdot w^* \geq \epsilon M_t \rightarrow \|w_t\| \geq \dfrac{\epsilon M_t}{\|w^*\|}$

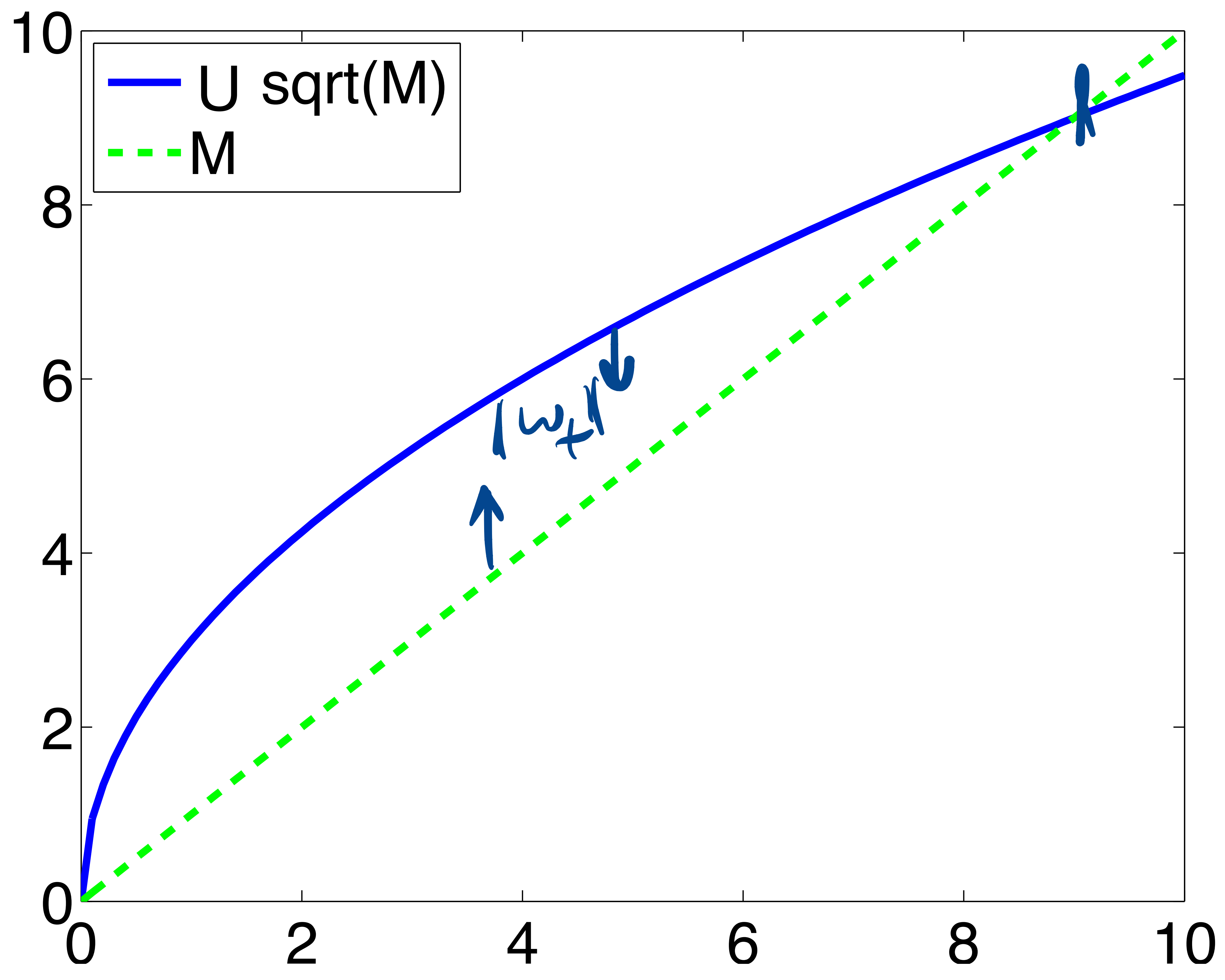
II $\quad w_t \cdot w_t \leq u^2 M_t \rightarrow \|w_t\| \leq u\sqrt{M_t}$

$\quad \|w_t\|^2$

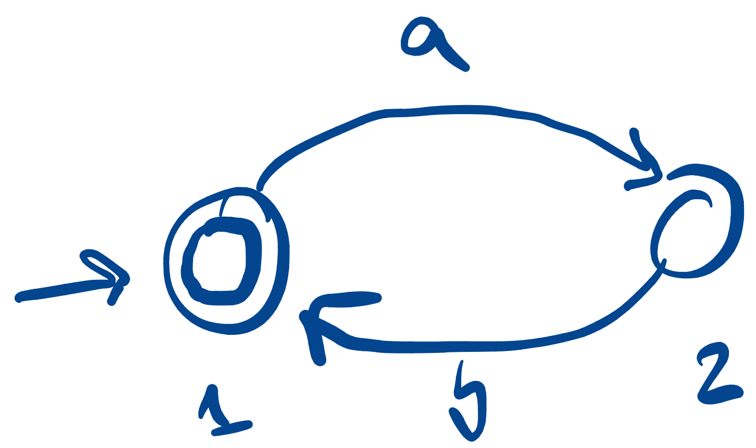
III $\quad M_t \leq \dfrac{u^2 \|w_*\|^2}{\epsilon^2}$

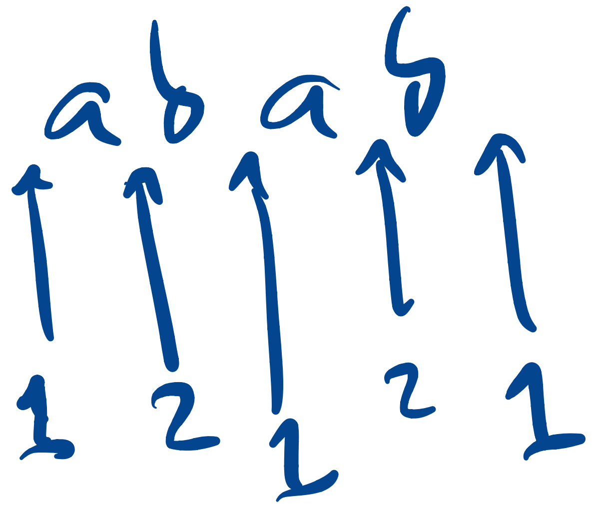


Upper and lower bounds



$(ab)^*$ $\quad \epsilon \quad ab \quad abab \quad \ldots$





assert: $\text{reachable}(1, t)$ $\quad$ $\text{final}(1)$

prove: $\exists s.\ \text{reachable}(s, \text{""}) \wedge \text{final}(s)$

assert: $\text{reachable}(1, t) \wedge \text{first}(t) = \text{"a"} \rightarrow \text{reachable}(2, \text{rest}(t))$

$\text{reachable}(2, t) \wedge \text{first}(t) = \text{"b"} \rightarrow \text{reachable}(1, \text{rest}(t))$

$$fib : \mathbb{N} \to \mathbb{N}$$

$$fib(0) = 1$$

$$fib(1) = 1$$

$$\forall x.\ x \geq 2 \to fib(x) = fib(x-1) + fib(x-2)$$

0	1	2	3	4	5	6
1	1	2	3	5	8	13

$$\mathrm{fib}(5) = \mathrm{fib}(4) + \mathrm{fib}(3)$$

$$= \mathrm{fib}(3) + \mathrm{fib}(2) + \mathrm{fib}(2) + \underbrace{\mathrm{fib}(1)}_{1}$$

def $\mathrm{fib}(n)$

$x \leq 1 \to 1 \mid$

$T \to \mathrm{fib}(x-1) + \mathrm{fib}(x-2)$

v

$x \leq 1 \Rightarrow v = 1$

$\neg(x \leq 1) \wedge T \to$

$v = \mathrm{fib}(x-1) + \mathrm{fib}(x-2)$

$\mathrm{fib} = \lambda x. \left(\quad \mathrm{fib} \quad \right)$

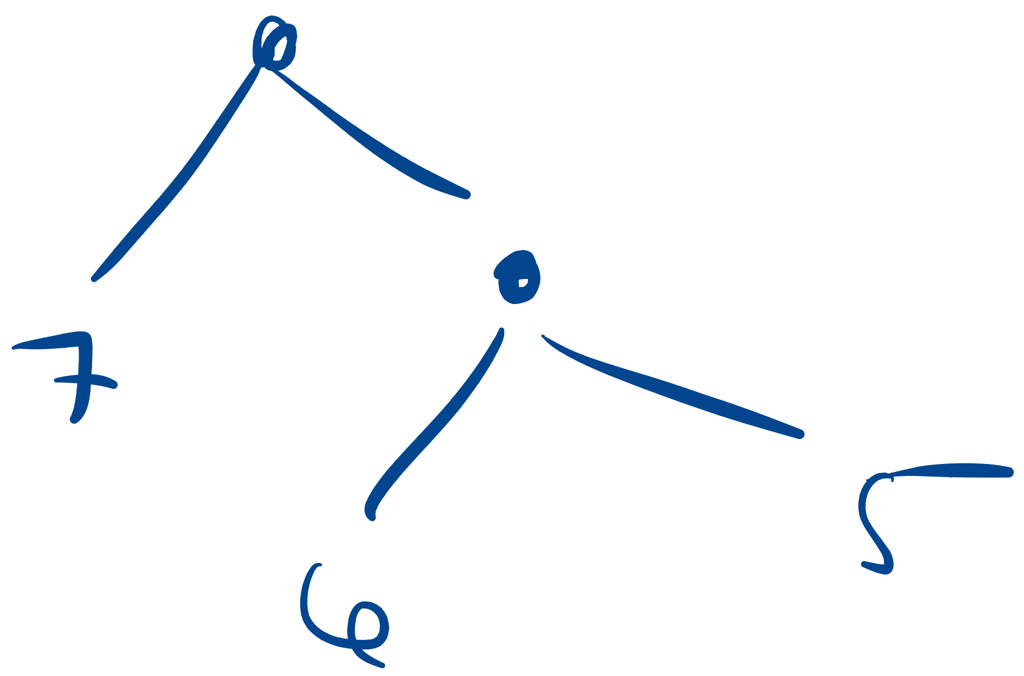
$a = 1 \qquad b = 1 \qquad i = 2$

$(a, b, i) \leftarrow (b, a+b, i+1)$

type string = empty() | cat(f: char, r: string)

type tree = leaf(v: int) | node(l: tree, r: tree)

node(leaf(7), node(leaf(6), leaf(5)))



λ node (l, r) . ..... l ... r

base case: for **every** base case constructor
prove $p(x)$ given x was constructed that way

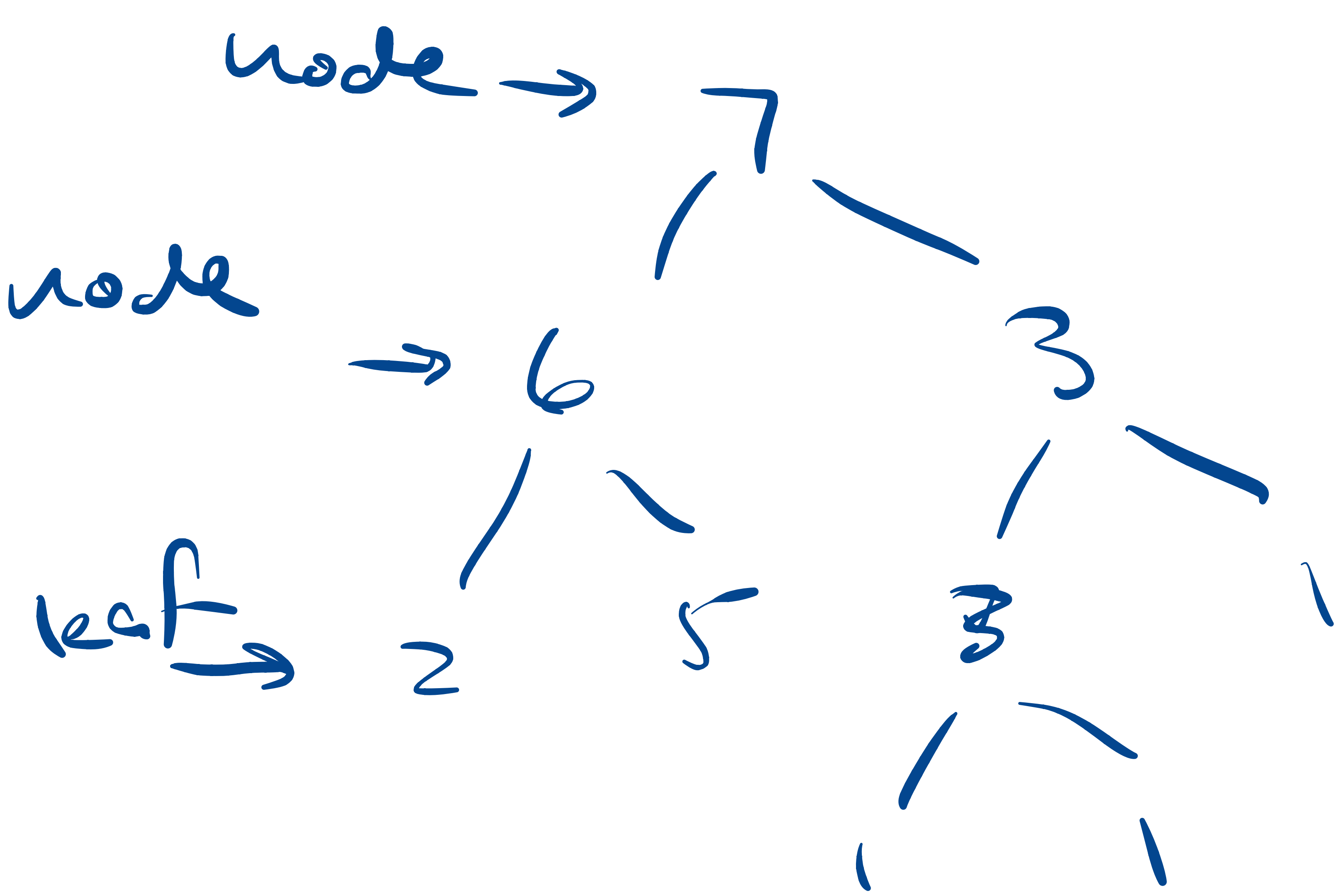
inductive case: for **every** inductive constructor
Assume inputs satisfy $p(x)$
prove output satisfies $p(x)$

type heap = leaf(v : int) | node(v : int, l : heap, r : heap)

$\forall x$: heap.

$(\lambda\, \text{leaf}(v).\ 0 \mid \lambda\, \text{node}(v, l, r).\ v - \max(\text{value}(l), \text{value}(r)))\ x$

≥ 0



def maxheap(x)

$[\lambda\, \text{leaf}(v).\ v \mid$

$$\lambda\, node(v, l, r).\ \max(v, \max(maxheap(l), maxheap(r)))$$

$$\exists x$$

$$\forall x : heap.\ \underbrace{value(x) = maxheap(x)}_{IH}$$

ind step: get $value(l) = maxheap(l)$

$value(r) = maxheap(r)$

$$maxheap(x) = \max(v, \max(value(l), value(r)))$$

$$= v$$

def value (x)

$$
\left(\begin{array}{l} \lambda\, \mathrm{leaf}(v).\ v \\ \lambda\, \mathrm{node}(v, l, r).\ v \end{array} \right| \Big) \; x
$$

def max (x, y)

$$
\left(\begin{array}{lcl} x \geq y & \to & x \\ T & \to & y \end{array} \right| \Big)
$$

type $\mathbb{N}$ = zero() | S(x: $\mathbb{N}$)

question after class about perceptron update rule