Tutorial IV

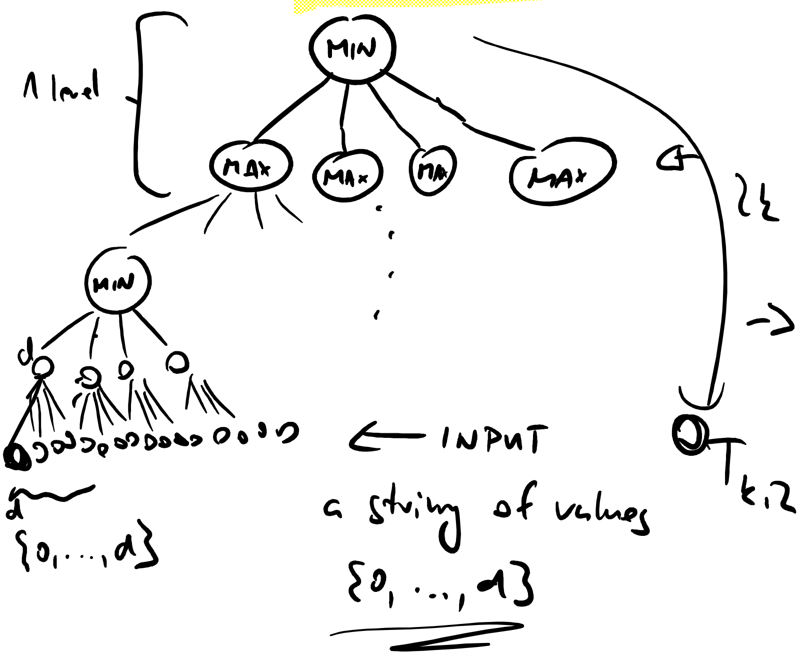
21 March 2019 12:00

GAME THEORETIC TECHNIQUES

↳ Game tree evaluation

↳ Yao's min-max technique for proving lower bounds

MIN-MAX TREES



$T_{k,D}$ - k levels

- d children

→ How many leaves? d^{2k}

$T_{k,2}$ $2^{2k} = 4^k$

Each leaf contains a value.

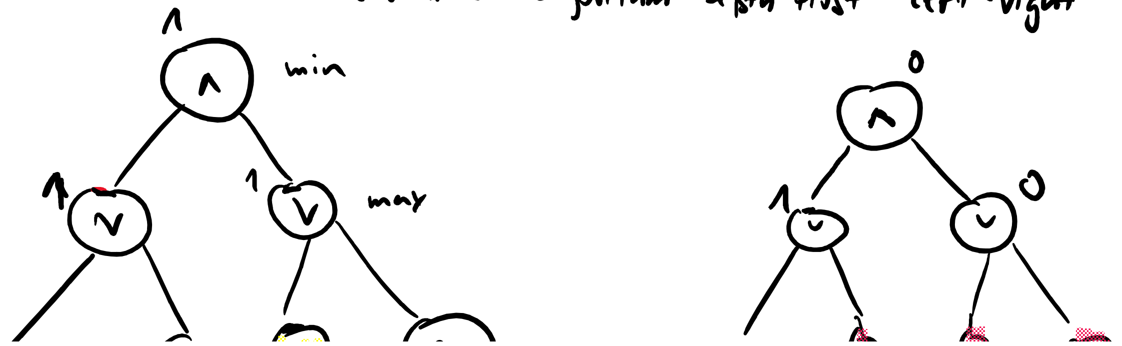
Each min node contains the smallest value of all it's children

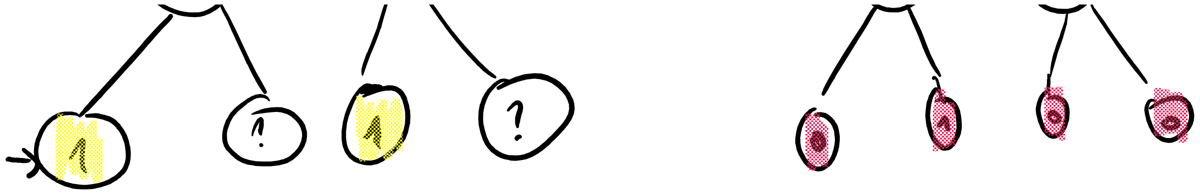
Each MAX node contains the largest value of all it's children

Goal is to find the value of the root.

$T_{k,2}$ ⇝ binary trees with input x a string $\{0,1\}^{4^k}$

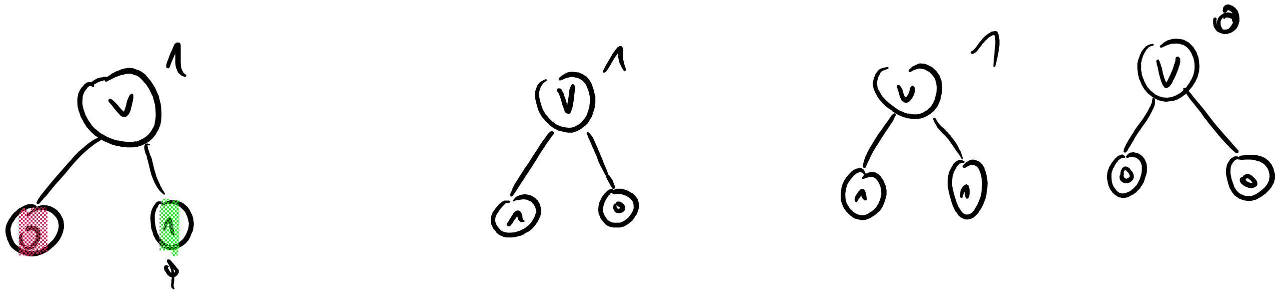
Deterministic algorithm: Depth-first left-right





Randomized algorithm.

Order of children evaluation is randomized.



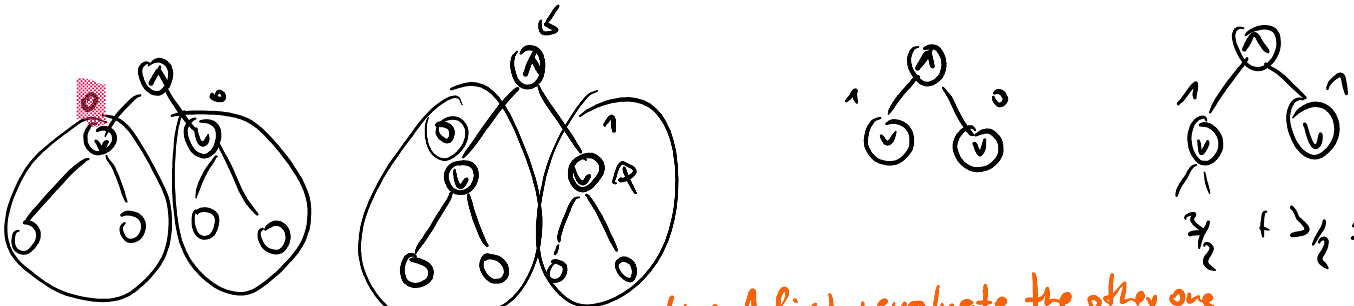
Red first = 2 evaluations | 3/2 evaluations | 1 evaluation | 2 evaluations

Green first = 1 evaluation

Average = 3/2 evaluations

det	2	1	1	2
rand	3/2	3/2	1	2

Induction $k \geq 1$



$\frac{3}{2} + \frac{3}{2} = 3$

2

case: find 0 first

$\frac{1}{2} \cdot 2 +$

case find 1 first + evaluate the other one

$\frac{1}{2} \cdot \left(\frac{3}{2} + 2\right)$

$1 + \frac{1}{2} \cdot \left(\frac{7}{2}\right)$

$< ?$ $< ?$

< 3 $\quad < 3 \quad 3$

$$E(x) = \frac{1}{2} \cdot E(\ldots) + \frac{1}{2} \cdot E(\ldots)$$

Overall average < 3

On average $T_{k,2}$ takes less than 3^k leaves to evaluate

I. H. $\quad T_{k-1,2}$ takes 3^{k-1} to evaluate

I. S. $\quad T_{k,2}$ takes 3^k.

$n = 2^{2k} \qquad 3^k = n^{0,79..}$

Basics of game theory

	R	P (Bob)	S
R	0	$\boxed{-1}$	1
P (Alice)	1	0	-1
S	-1	1	0

← Strategies

strategies

$O_A = -1$

$O_B = 1$

→ Game evaluation matrix

→ Alice is trying to maximize the outcome

→ Bob is trying to minimize the outcome

Generally this Matrix is $[M_{ij}]$ of real numbers M_{ij}

if Alice chooses strategy i, in the worst case if Alice chooses strategy i, in the worst case she gets $\min_j M_{ij}$.

What is Alice's best choice? $\boxed{\max_i \min_j M_{ij}} = O_A$

What is Bob's best choice? $\boxed{\min_j \max_i M_{ij}} = O_B$

There are games, for which $O_A = O_B$

$$\begin{matrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & \boxed{0} \end{matrix}$$

$O_A = 0$

$O_B = 0$

MIXED strategies

Alice = probability distribution on rows p

Bob = probability distribution on columns q

column vectors of probability

$$p^T M q = \sum_i \sum_j p_i q_j M_{ij}$$ = Average value of a game with mixed strategies p and q.

for fixed strategy of Alice p, she is guaranteed to get at least $\min_q p^T M q$ points

Alice's best strategy is $\max_p \min_q p^T M q$

Bob's best strategy is $\min_q \max_p p^T M q$

Von Neumann's thm.

$$\max_p \min_q p^T M q = \min_q \max_p p^T M q$$

Loomis thm

$$\max_p \min_k p^T M e_k = \min_q \max_j e_j^T M q,$$

where $e_i = (0, \dots 1, \dots 0)$ i^{th} position

Proof.

for fixed p $p^T M q$ is a function linear in $q_1, \dots, q_n$

scalars variables

$$p^T M q = a_1 q_1 + a_2 q_2 + \dots + a_n q_n$$

$$q_1 + q_2 + \dots + q_n = 1$$

find smallest a (wlog a_k)

and min is obtained by $q = (0 \ldots, 1, \ldots 0)$

k^{th} position

	A_1	A_2	.....	A_n
I_1	$c(I_1, A_1)$			
I_2				
I_3				
$\vdots$				
I_n				

$C(I_j, A_k)$ is the length of computation of A_k on input I_j.

$E[C(I_p, A_q)]$ = expected running time for input distribution p and rand. algorithm specified by q.

vNeumann's thm:

$$\max_p \min_q E(C(I_p, A_q)) = \min_q \max_p E(C(I_p, A_q))$$

Loomis thm:

$$\max_{p} \min_{A_i \in \mathcal{A}} E(C(I_p, A_i)) = \cancel{\min_{q}} \max_{i \in I} E(C(I_i, A_q))$$

the worst average for a problem

$$\boxed{\forall_p}, q \qquad \boxed{\min_{A_i \in \mathcal{A}} E(C(I_p, A))} \le \max_{i \in I} E[C(I_i, A_q)]$$

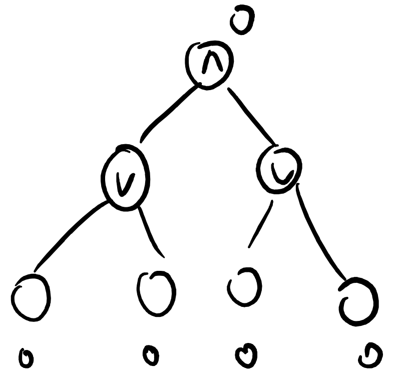
it holds for all A_q

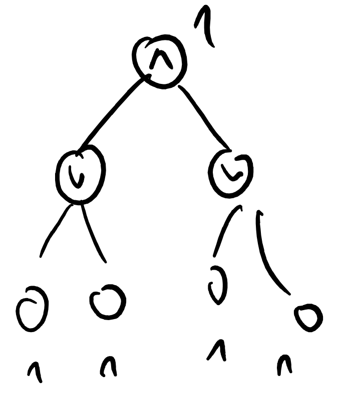
We need to find distribution on the inputs which has an optimal deterministic algorithm, with complexity as high as possible.

Examples

Input for tree evaluation is always all 0 or always all 1 vector

What is the optimal det. algorithm?



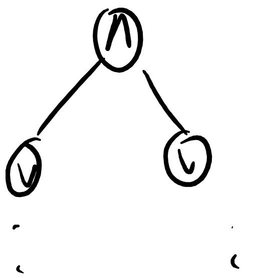


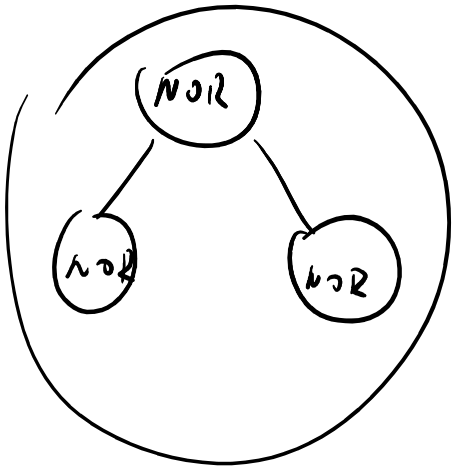
$\boxed{1 \text{ step}}$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & \to & 0 \\ 0 & 0 & 1 & 1 & \to & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 1 & 0 & \to & 1 \\ 1 & 0 & 0 & 1 & \to & 1 \end{bmatrix}$$

$\boxed{\text{2 steps}}$





$(a \vee b) \wedge (c \vee d) = (a \text{ NOR } b) \text{ NOR } (c \text{ NOR } d)$

0 0

$$\boxed{\begin{array}{l} \Pr(\text{leaf}=1) = p \\ \Pr(\text{leaf}=0) = 1-p \end{array}}$$

0 0	1
0 1	0
1 0	0
1 1	0

$n^{0.6\ldots} \leq n^{0.79\ldots}$

$n^{0.75\ldots} = n^{0.79}$

$(1-p)^2 = p$

$p = \frac{(3-\sqrt{5})}{2}$