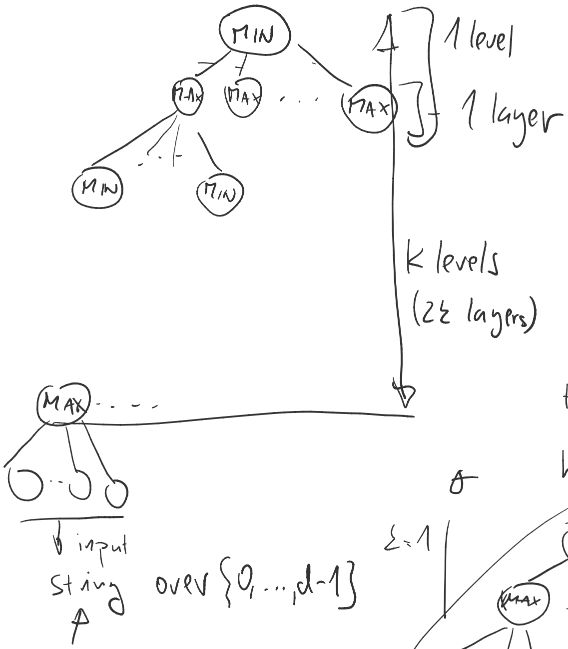
Tutorial IV

25 March 2021 15:57

Game tree evaluation

MIN MAX TREES (DEFINITION)



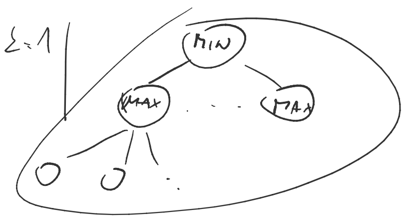
$T_{k,d}$ tree

d-children

k-levels

2k+1 layers

How many leaves does a $T_{k,d}$ tree have? d^{2k}



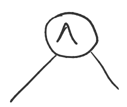
- → Each leaf contains a value
- → Each MIN node contains the smallest of it's childrens values
- → Each MAX node contains the largest value of it's children's values.
- → GOAL is to find the value of the root

$T_{k,2}$ - Binary trees with input $2^{2k} = 4^k$ bits

min → $\wedge$

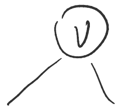
max → $\vee$

min



1 1 1

0 1 0

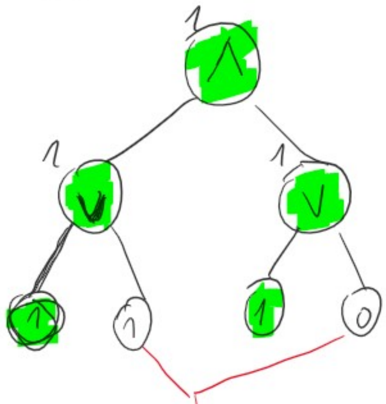


max

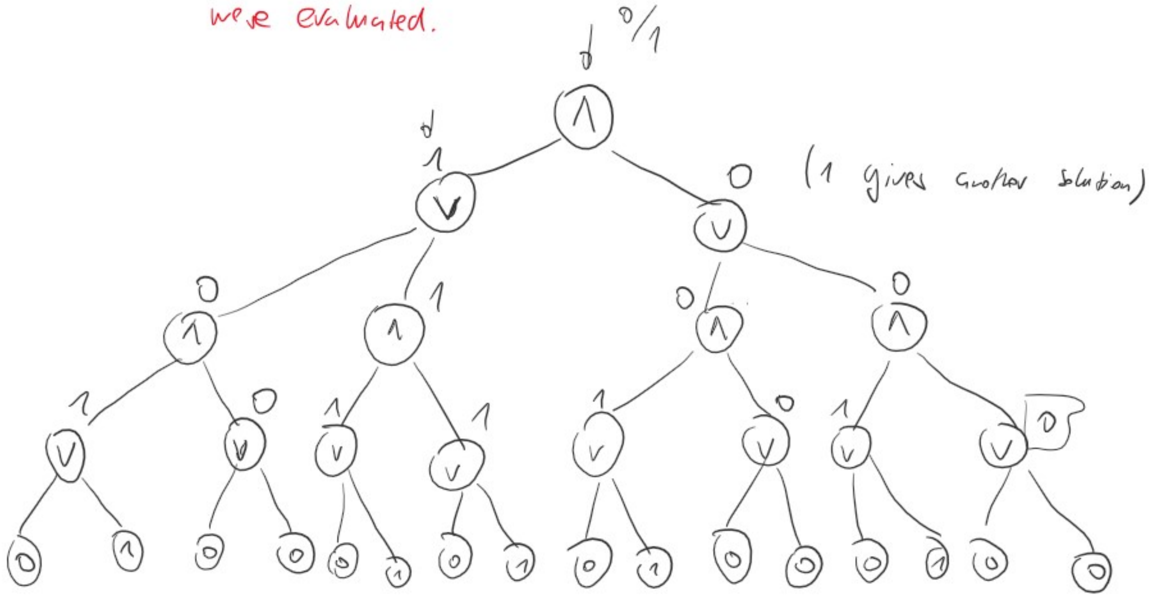
0 0 0

0 1 1

Deterministic evaluation algorithm ('depth first', 'left child first')



Only 4 internal nodes were evaluated.



All ~32 leaves need to be accessed

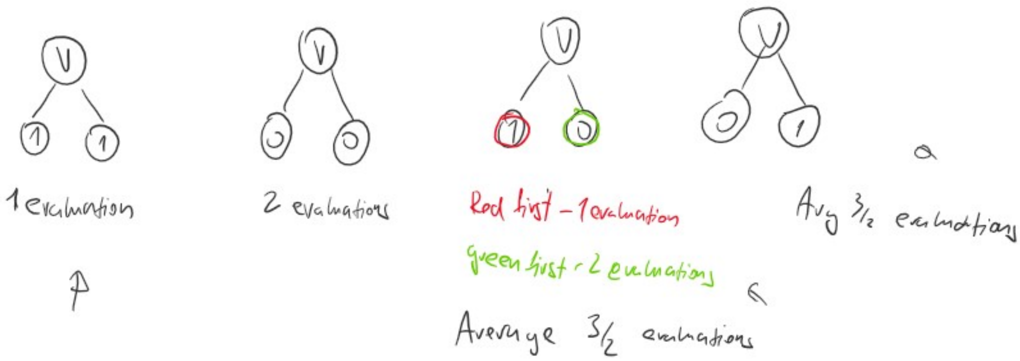
⇒ the difficulty of any deterministic algorithm is $O(4^k)$ (worst case)

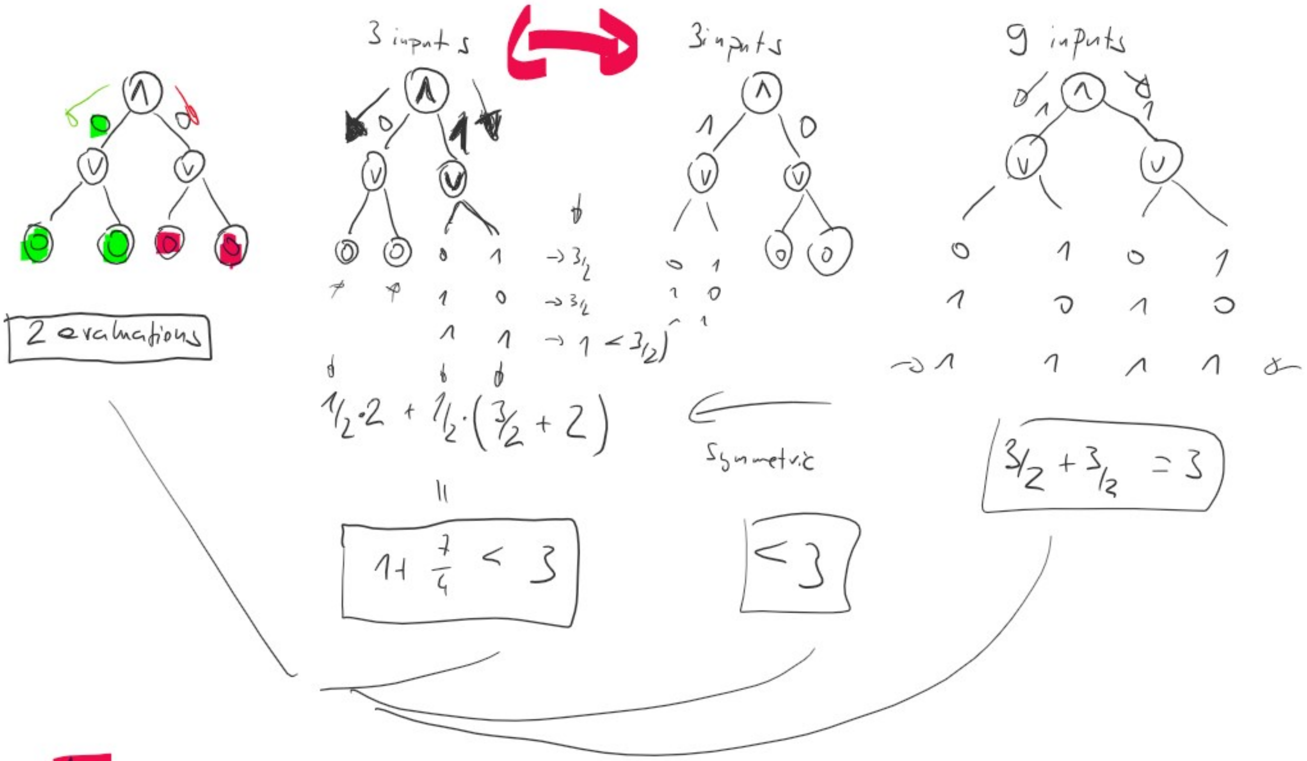
$n = 4^k$

Randomized algorithm – choose the child to evaluate at random

Claim: Expected complexity is $O(3^k)$ $n^{0.75..}$

Proof:





Avg Evaluations ≤ 3

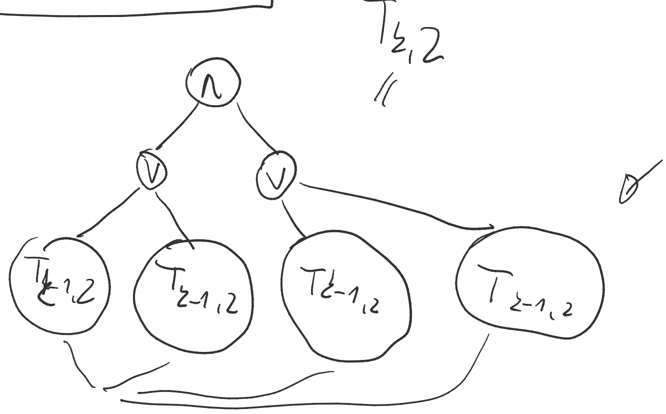
Proof by induction (number of levels)

k = 1 Basis case

Induction hypothesis

$T_{k-1,2}$ can be evaluated with randomized algorithm by accessing less than 3^{k-1} leaves on average.

Induction step



Each of these takes at most 3^{k-1} leaf evaluations (I.H.)

By I.B., we need to evaluate 3 of $T_{k-1,2}$ subtrees on average. $\Rightarrow$ $T_{k,2}$ needs at most $3 \cdot 3^{k-1} = 3^k$ leaf evaluations.