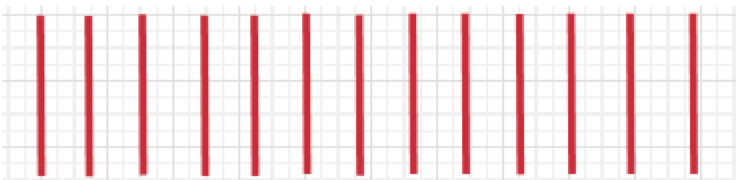
03/04/2020 **_ECE 479/579 Module_5_**

Searching Game trees

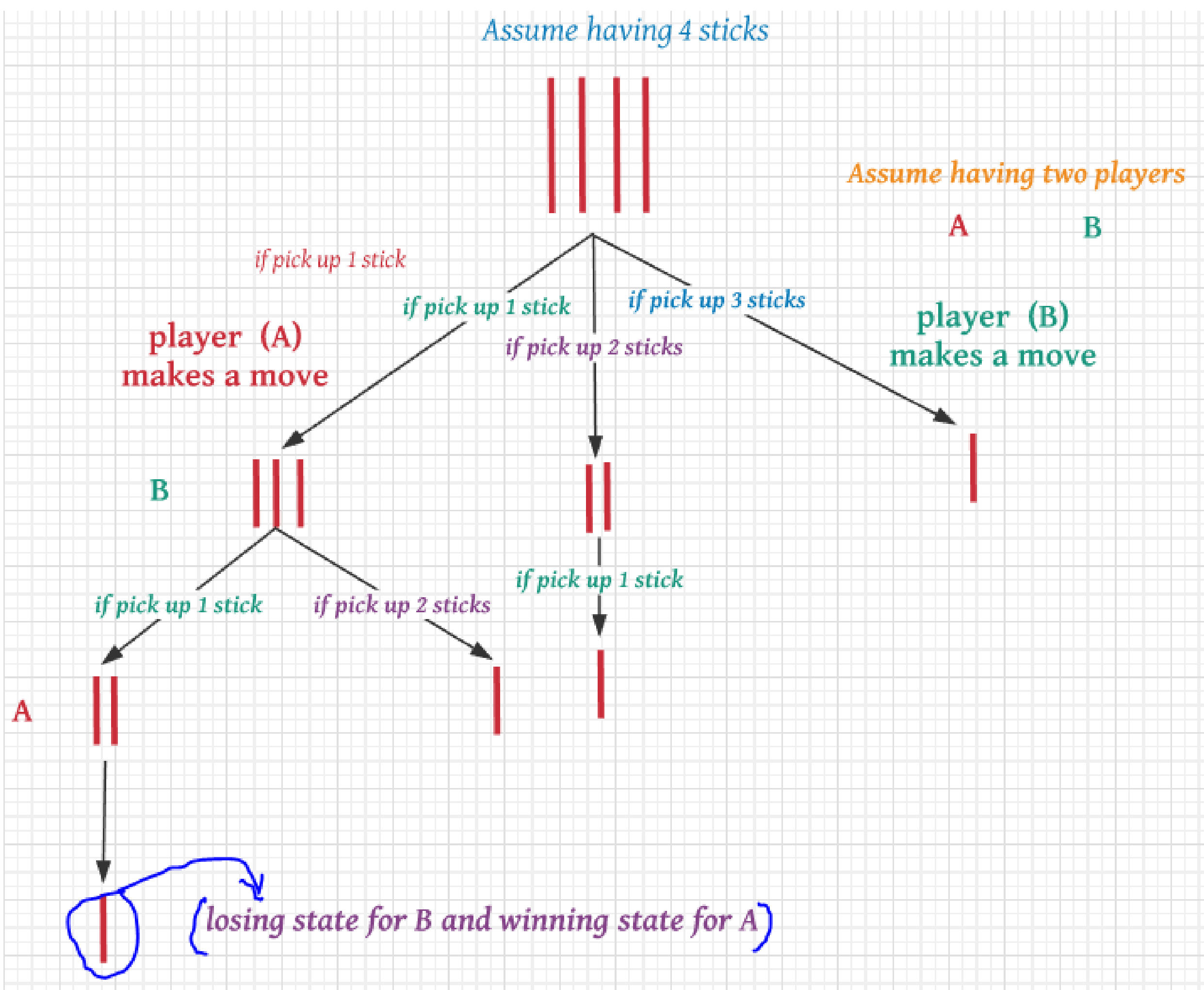
- *Represent a situation where is two parities involve*

Game: assume having 13 sticks as showing



Game rules

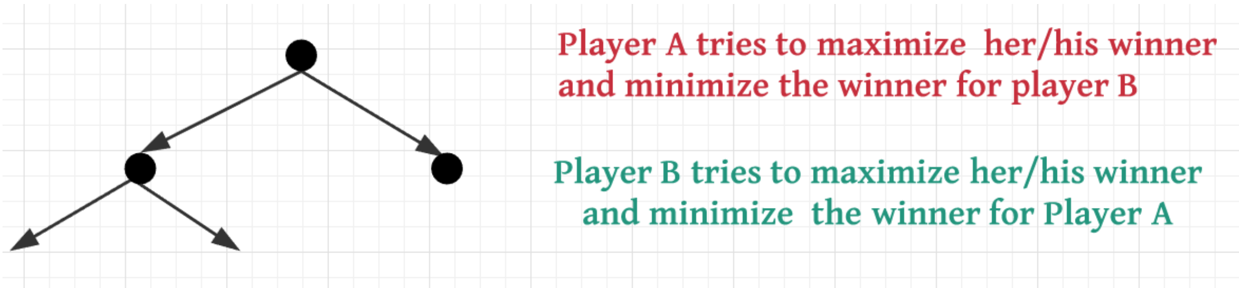
- *Player can take 1, 2, or 3 sticks onetime*
- *The player who takes the last sticks loses the game*

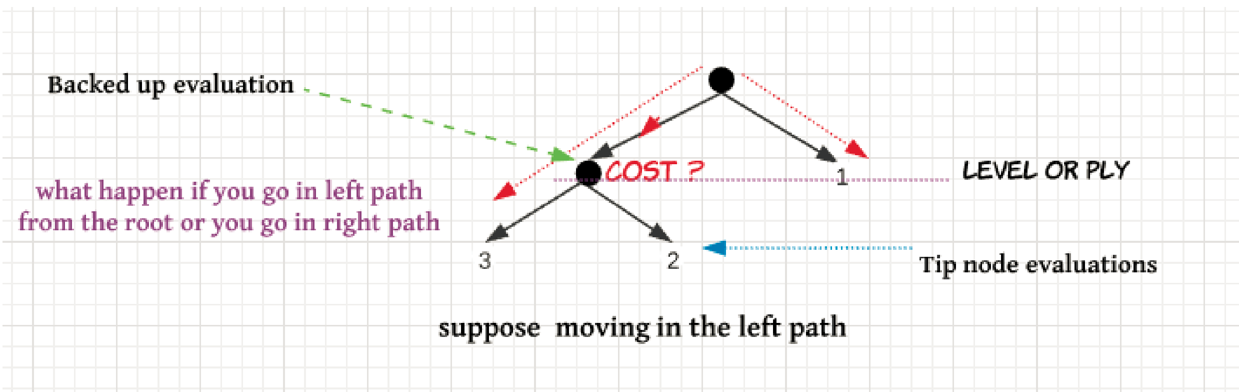


Search Game tree

Basic concept

1. *Min /Max* — *Maximizing player (Max)* / *Minimizing player (Min)*





There are two types of evaluation associated with node

1. *Tip node evaluation (reward)*
2. *Backed up evaluation. (internal node of the tree)*

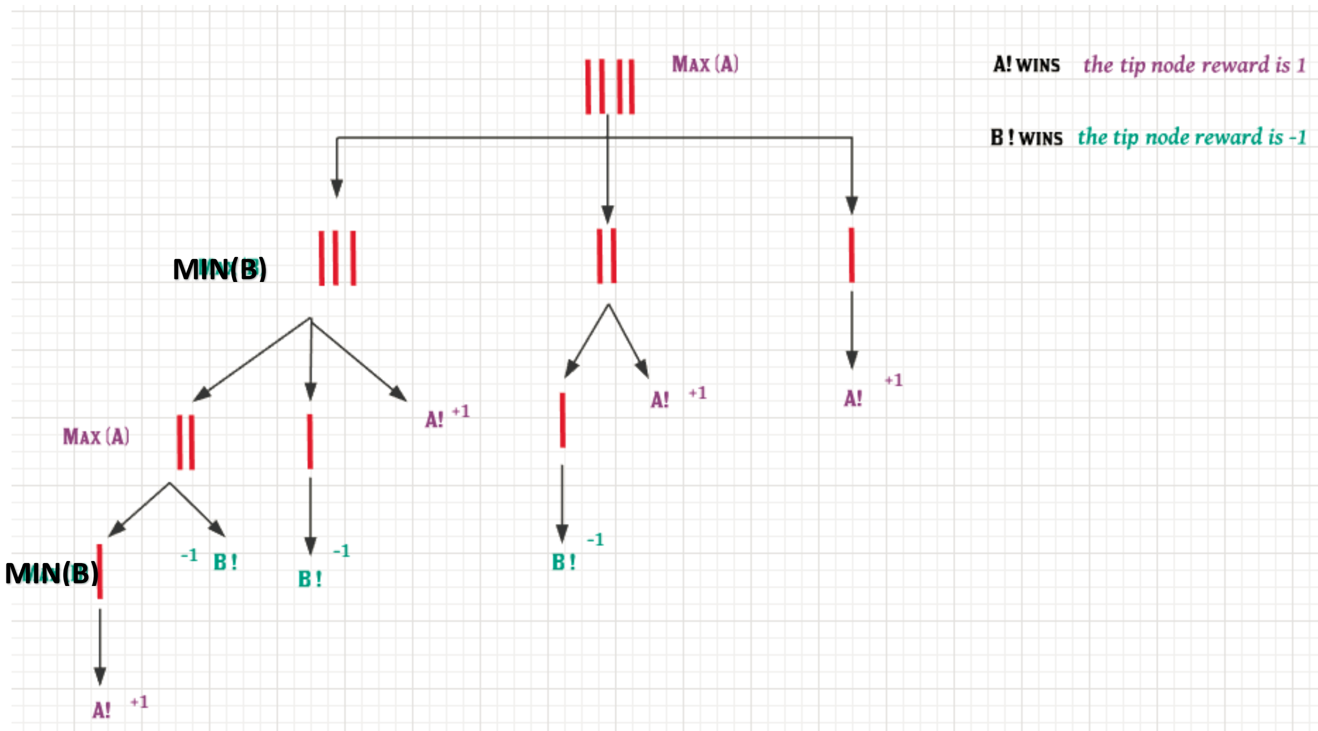
To compute tip and backed up

- *Use the min/max if knowing the tip node evaluation.*

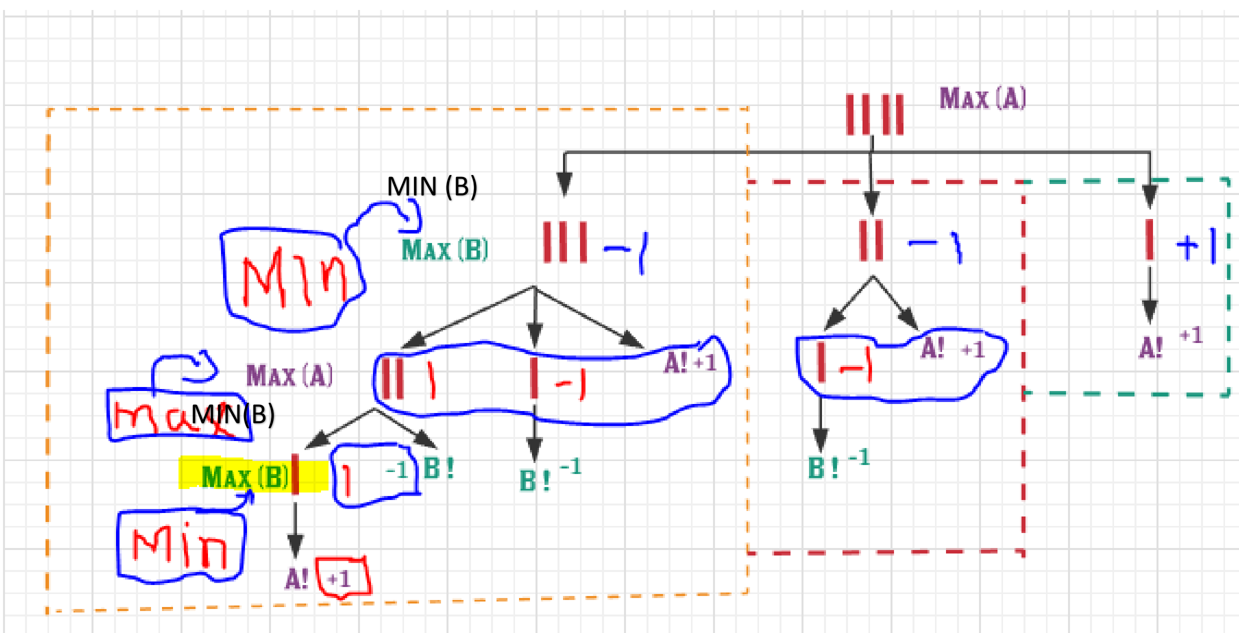
Alpha beta algorithm is about Reduce the number of evaluation

- *For the max node back up value is the max of children values*
- *For the min nodes backup value is the min of children values*

Example

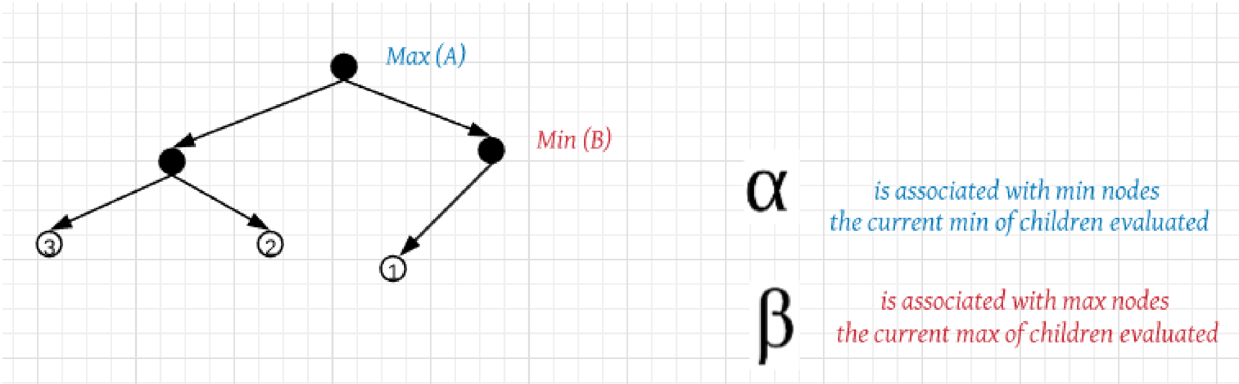


Compute backed up value by using Min/Max

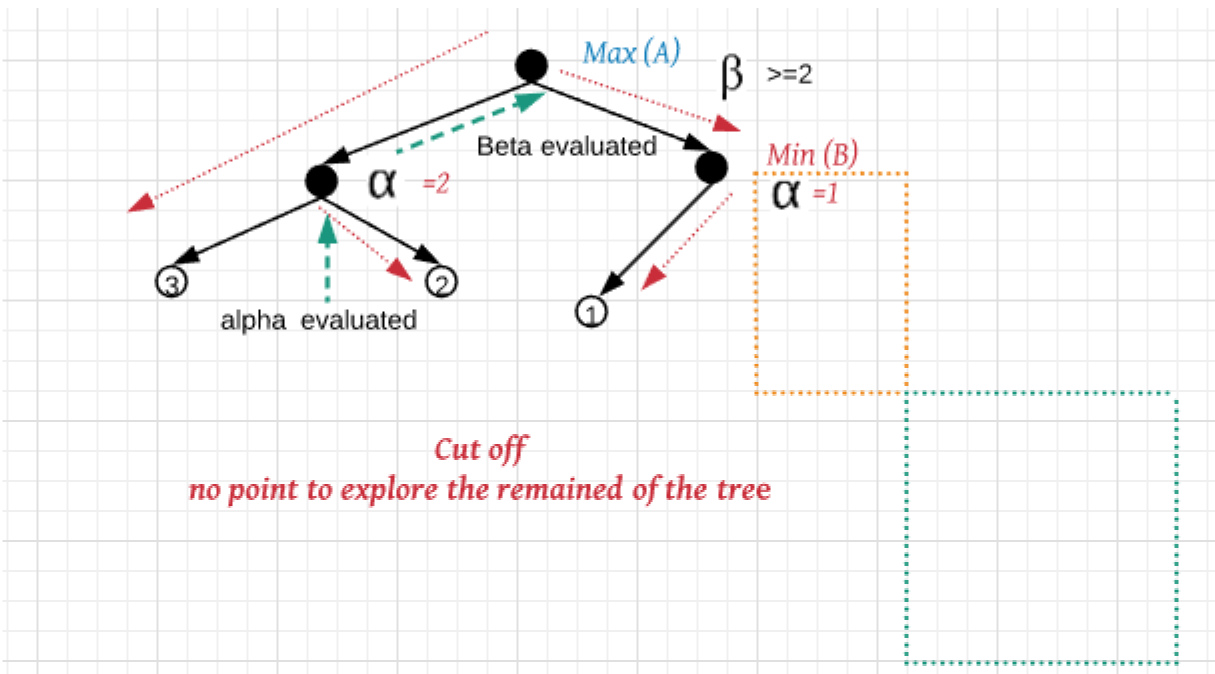


What is good about it and what is bad about it?

It is good about the computation complex problem (game of checker and chess)

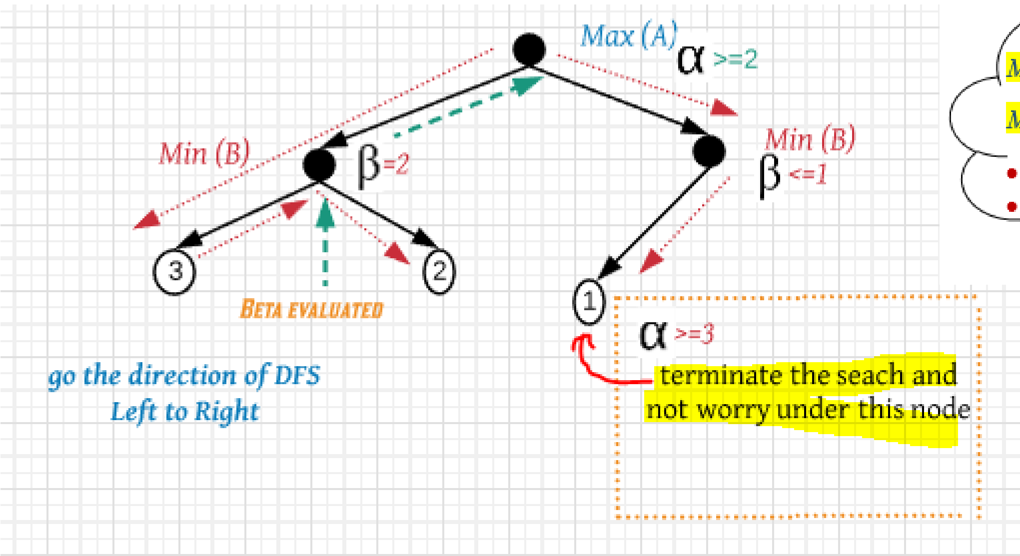


Continue exploring search tree DFS left first. Always do in DFS first and left to right direction



Continue with game tree

Alpha and beta pruning (search algorithm) $\alpha\beta$



Correction!!!

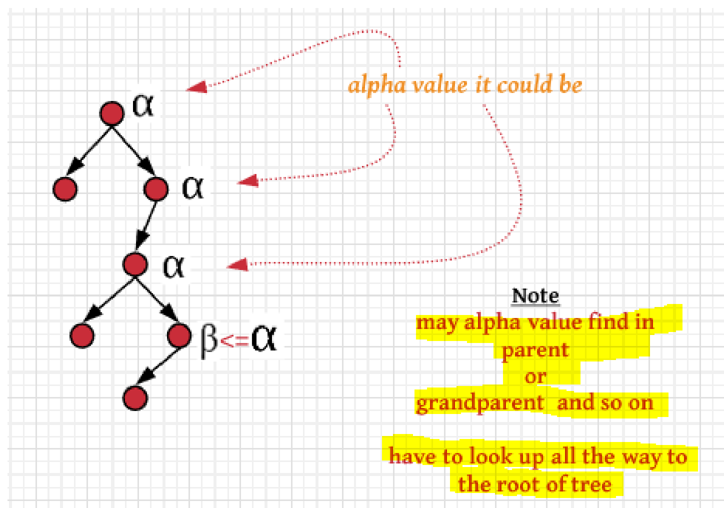
Max: α value (max of the children values)

Min: β value (min of children values)

- *The alpha vale never decrease*
- *The Beta value never increase*

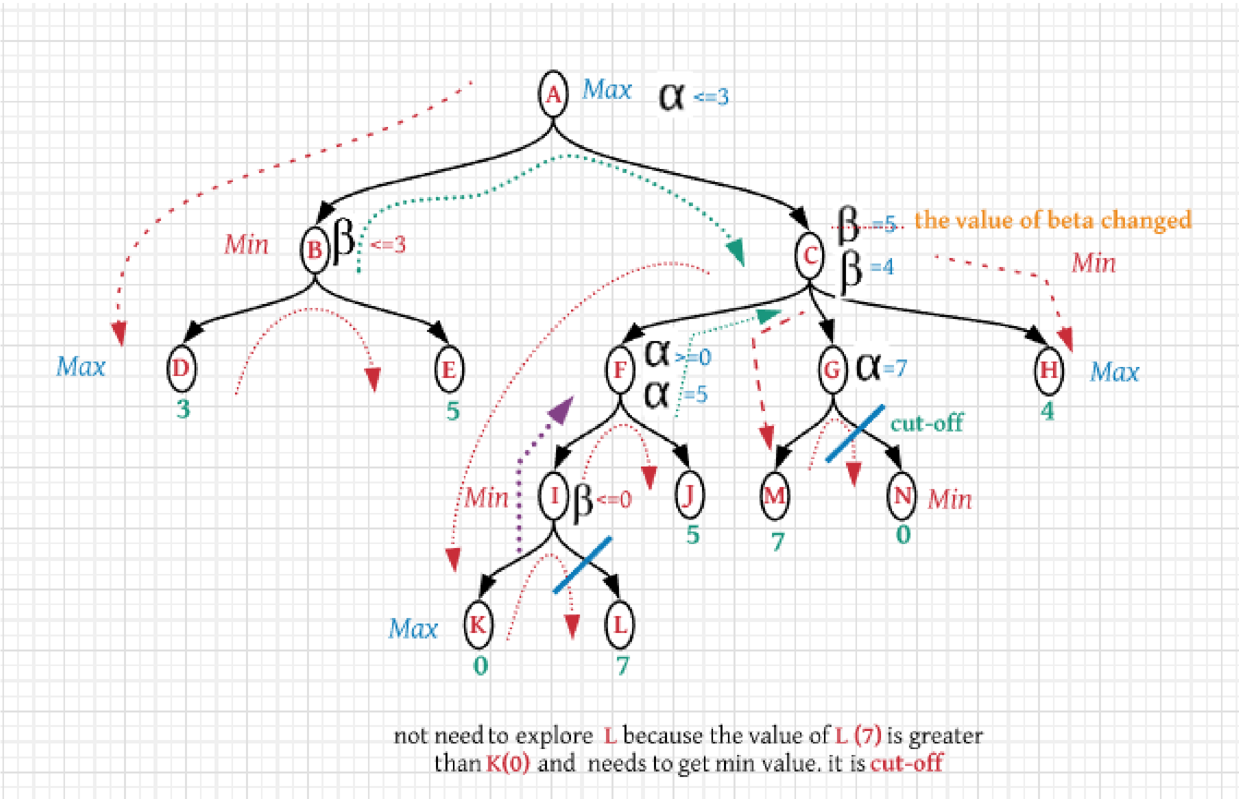
α/β pruning

1. Searching can be discontinued below any Min node whose $\beta <= \alpha$ of any of its ancestors. (alpha cut-off)



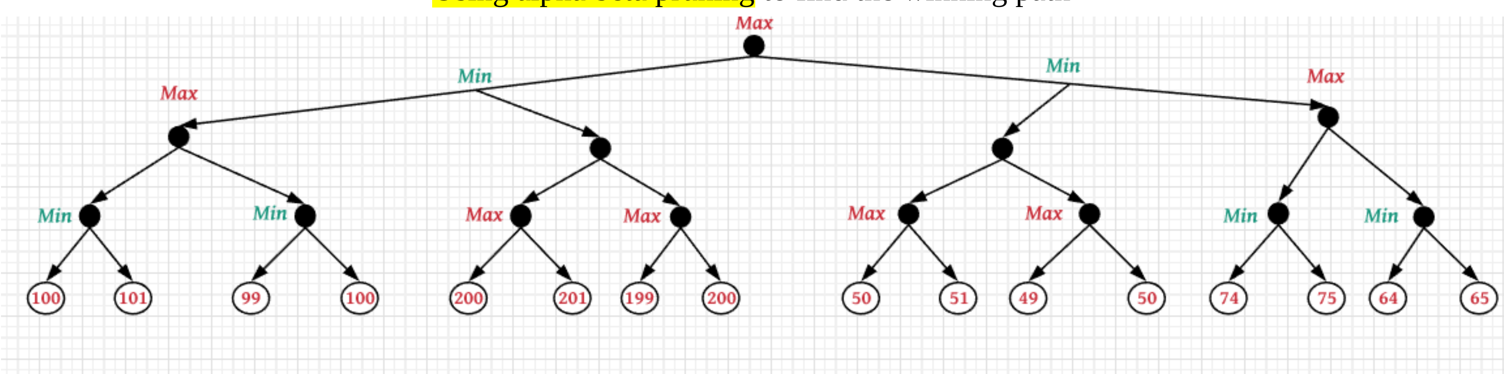
2. Search can be discontinued below an Max node whose $\alpha >= \beta$ of any of its ancestors (beta cut-off)

Example



Example (exercise)

Using alpha beta pruning to find the winning path

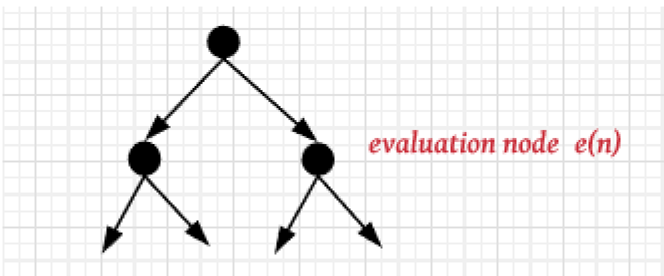


What is the limitation?

If have 16 parallel processors, it is easy to evaluate by using alpha beta.

<u>Example</u>

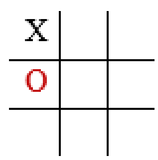
To demonstrate that partial state of the game can be assigned an evaluation to help alpha beta pruning.



<u>Example</u>

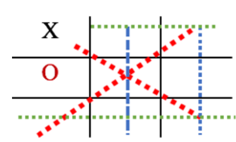
Tic Tac Toe game

<u>State</u>

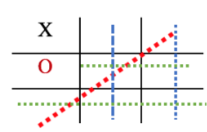


- X for Max player
- O for Min player
- e(s) if S is a win for max
  then e(s) =∞
- e(s) if s is a win for min
  then e(s) = - ∞

  e(s) = {number of rows, columns, and diagonal open for max}
  e(s) = - {number of rows, columns, and diagonal open for min}



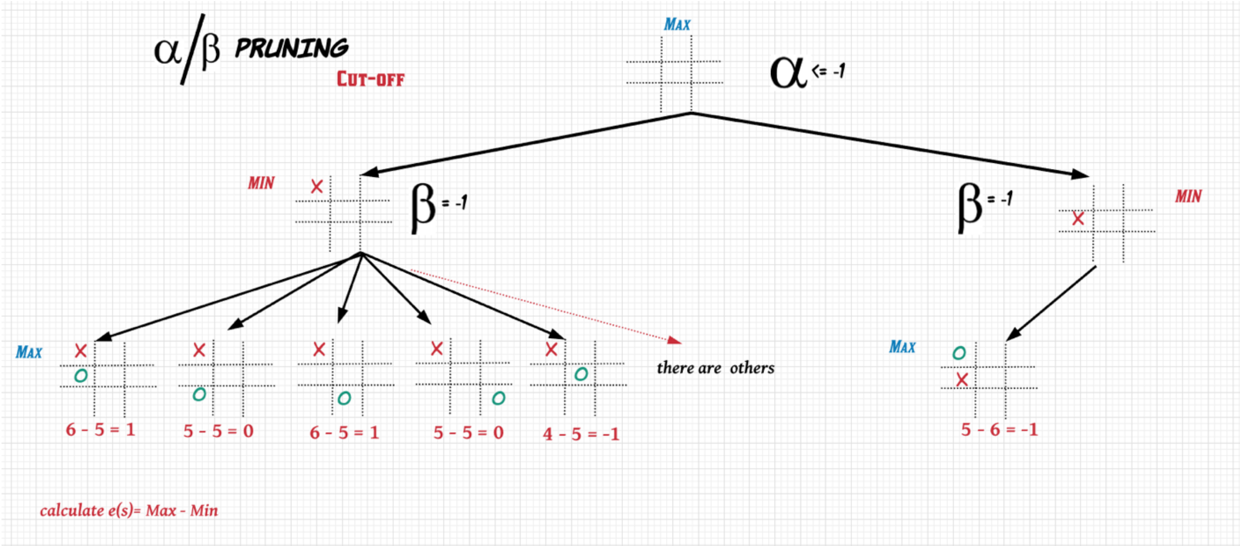
- calculate the number of rows, column and diagonal for X
  2 rows, 2 columns, 2 diagonals. → 2+2+2 = 6



- calculate the number of rows, column and diagonal for O
  2 rows, 2 columns, 1 diagonal. → 2+2+1 = 5

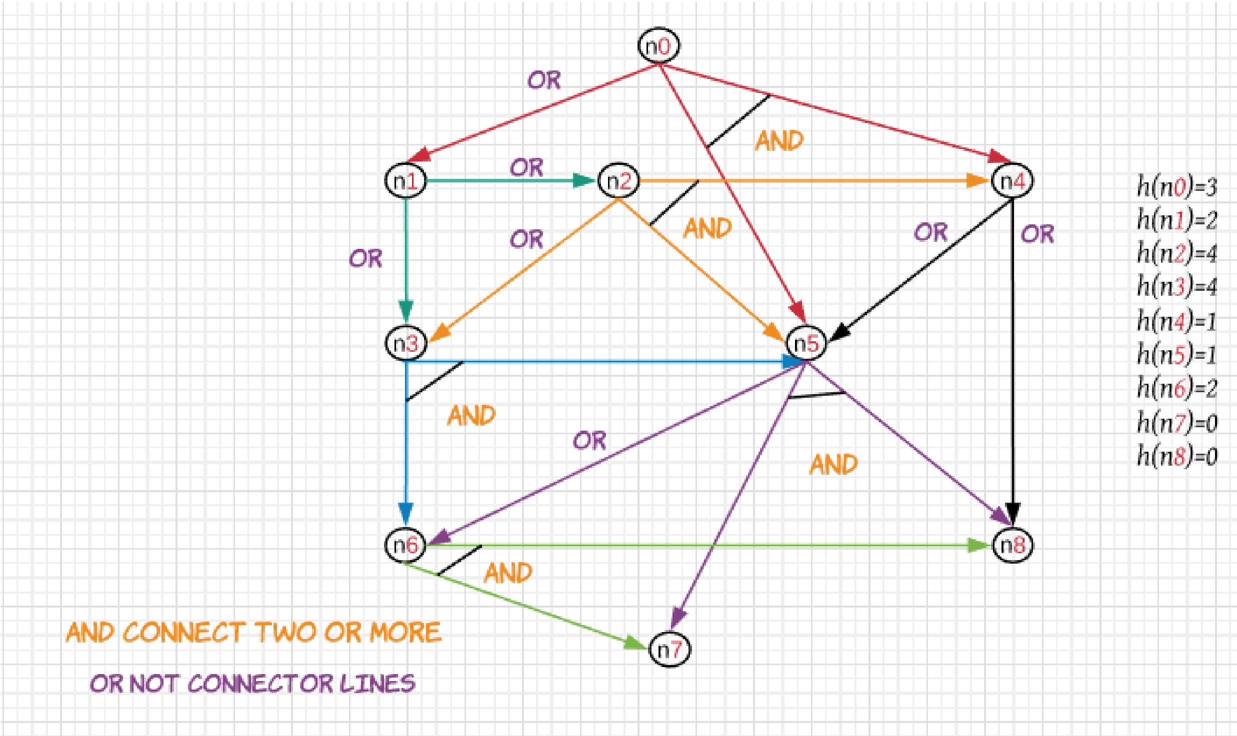
  e(s)= Max – Min = 6-5= 1

Example



α/β PRUNING
CUT-OFF
MAX
α <= -1
MIN
β = -1
β = -1
MIN
MAX
there are others
MAX
6 - 5 = 1
5 - 5 = 0
6 - 5 = 1
5 - 5 = 0
4 - 5 = -1
5 - 6 = -1
calculate e(s)= Max - Min

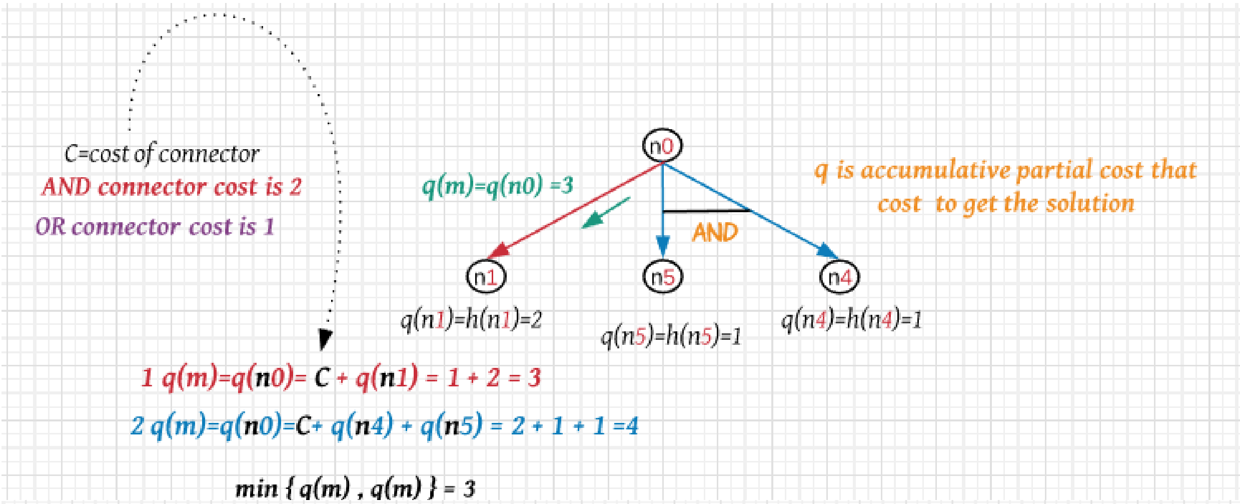
AO*

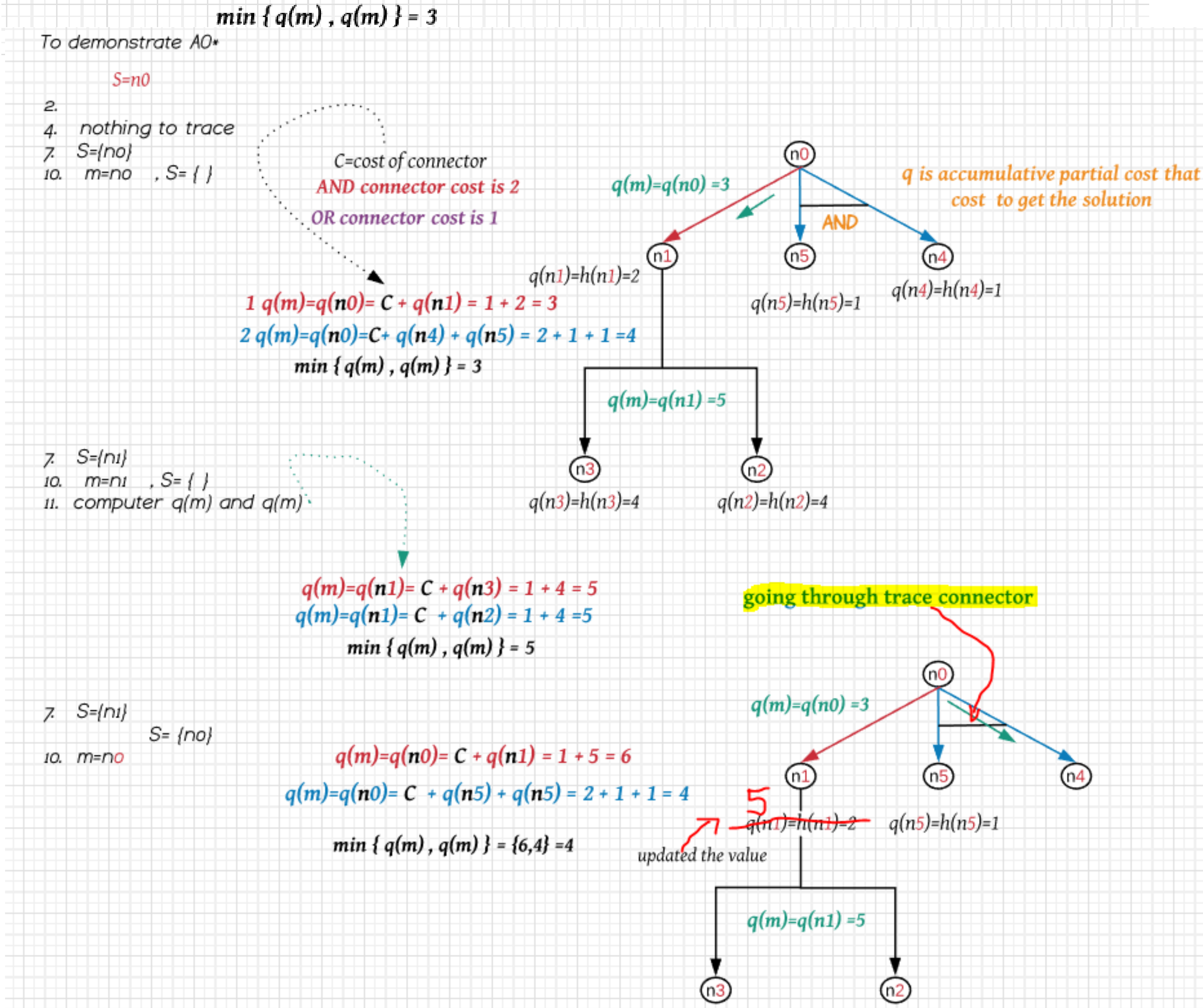


- *Algorithm has two loops*
- *Loops keep building partial solution while building, it is evaluated what it the best subgraph so far.*
- *Check to see if by any chances, it might be better solution that has better partial cost so far, if that the cases switch to pursue that particular solution.*

- *Circle node n7 and node n8 with green*
- *the green color indicates something solve*

Example


C=cost of connector
AND connector cost is 2
OR connector cost is 1
n0
q(m)=q(n0) =3
q is accumulative partial cost that cost to get the solution
AND
n1
n5
n4
q(n1)=h(n1)=2
q(n5)=h(n5)=1
q(n4)=h(n4)=1
1 q(m)=q(n0)= C + q(n1) = 1 + 2 = 3
2 q(m)=q(n0)=C+ q(n4) + q(n5) = 2 + 1 + 1 =4
min { q(m) , q(m) } = 3


To demonstrate AO*
S=n0
2.
4. nothing to trace
7. S={no}
10. m=no , S= { }
C=cost of connector
AND connector cost is 2
OR connector cost is 1
n0
q(m)=q(n0) =3
q is accumulative partial cost that cost to get the solution
AND
n1
n5
n4
q(n1)=h(n1)=2
q(n5)=h(n5)=1
q(n4)=h(n4)=1
1 q(m)=q(n0)= C + q(n1) = 1 + 2 = 3
2 q(m)=q(n0)=C+ q(n4) + q(n5) = 2 + 1 + 1 =4
min { q(m) , q(m) } = 3
q(m)=q(n1) =5
7. S={n1}
10. m=n1 , S= { }
11. computer q(m) and q(m)
n3
n2
q(n3)=h(n3)=4
q(n2)=h(n2)=4
q(m)=q(n1)= C + q(n3) = 1 + 4 = 5
q(m)=q(n1)= C + q(n2) = 1 + 4 =5
min { q(m) , q(m) } = 5
going through trace connector
7. S={n1}
S= {no}
10. m=no
n0
q(m)=q(n0) =3
q(m)=q(n0)= C + q(n1) = 1 + 5 = 6
q(m)=q(n0)= C + q(n5) + q(n5) = 2 + 1 + 1 = 4
min { q(m) , q(m) } = {6,4} =4
n1
n5
n4
5
~~q(n1)=h(n1)=2~~
q(n5)=h(n5)=1
updated the value
q(m)=q(n1) =5
n3
n2