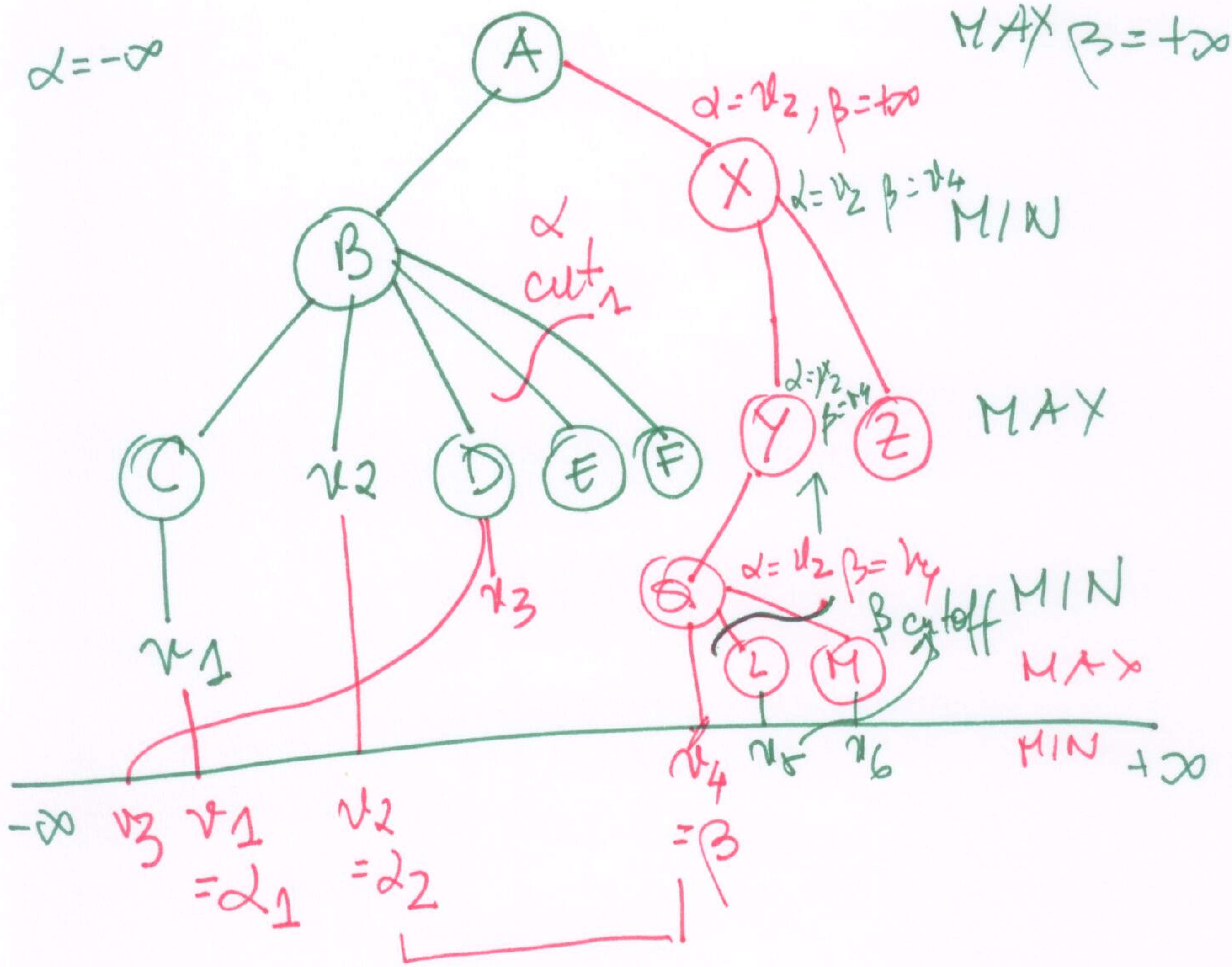


Discussing α-β Prunning

function **ALPHABETA-SEARCH** (*state*)

returns *an action*

Inputs: *state,* current state in game

$v \leftarrow$ MAX-VALUE(*state*, $-\infty$, $+\infty$)

return the *action in SUCCESSORS*[*state*] with value *v*

---

function MAX-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow -\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MAX(*v*, MIN-VALUE(*s*, α, β))

if $v \geq \beta$ **then return** *v*

$\alpha \leftarrow$ MAX (α, *v*)

return *v*

---

function MIN-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow +\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MIN(*v*, MAX-VALUE(*s*, α, β))

if $v \leq \alpha$ **then return** *v*

$\beta \leftarrow$ MIN(β, *v*)

return v

Remember when you do α-β Prunning

1. You can change the value of α only in a MAX node.

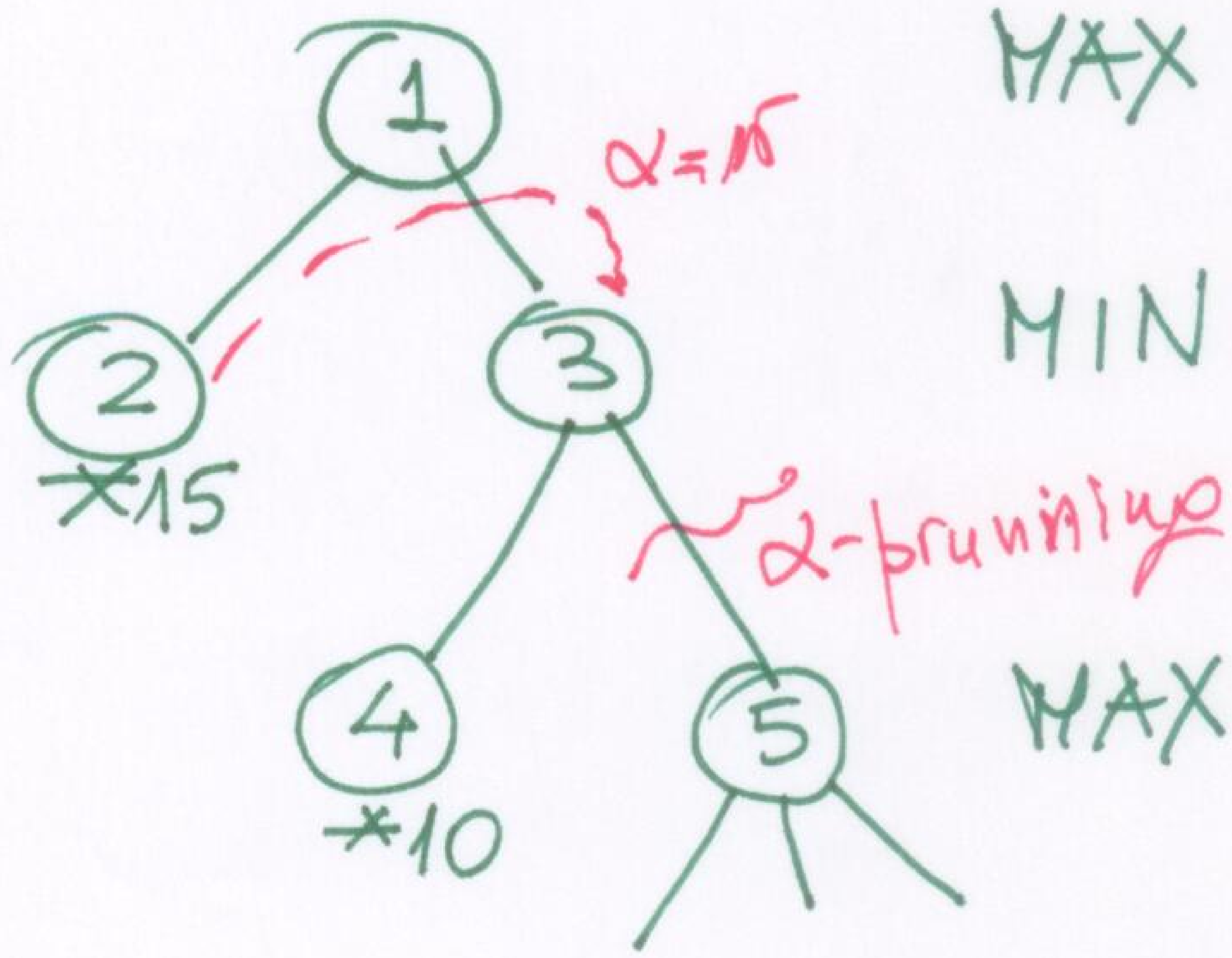
2. You can change the value of β only in a MIN node.

You can prune the game tree in 2 ways:

(1) if in a MAX node, (max-value $\geq \beta$) $\Rightarrow$ β-prunning of remaining successors

(2) if in a MIN node, (min-value $\leq \alpha$) $\Rightarrow$ α-prunning of remaining successors

Example of prunning #1



function **ALPHABETA-SEARCH** (*state*)

returns *an action*

Inputs: *state,* current state in game

$v \leftarrow$ MAX-VALUE(*state*, $-\infty$, $+\infty$)

return the *action in SUCCESSORS*[*state*] with value v

---

function MAX-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow -\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MAX(v, MIN-VALUE(s, α, β))

if $v \geq \beta$ **then return** v

$\alpha \leftarrow$ MAX (α, v)

return v

---

function MIN-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow +\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MIN(v, MAX-VALUE(s, α, β))

if $v \leq \alpha$ **then return** v

$\beta \leftarrow$ MIN(β, v)

return v

State	α	β	v	Function
1	$-\infty$	$+\infty$	$-\infty$	Max-Value
2	return		15	Min-Value
1	15	$+\infty$	15	Max-Value
3	15	$+\infty$	$+\infty$	Min-Value
4	return		10	Max-Value
3	15	$+\infty$	10	Min-Value
1	15	$+\infty$	15	Max-Value
no other successors $\Rightarrow$	15	15	15	

α-prunning of the remaining successors of state 3

Example of cut 2

function **ALPHABETA-SEARCH** (*state*)

returns *an action*

Inputs: *state,* current state in game

$v \leftarrow$ MAX-VALUE(*state*, $-\infty$, $+\infty$)

return the *action in SUCCESSORS*[*state*] with value v

---

function MAX-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow -\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MAX(v, MIN-VALUE(s, α, β))

if $v \geq \beta$ **then return** v

$\alpha \leftarrow$ MAX (α, v)

return v

---

function MIN-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow +\infty$

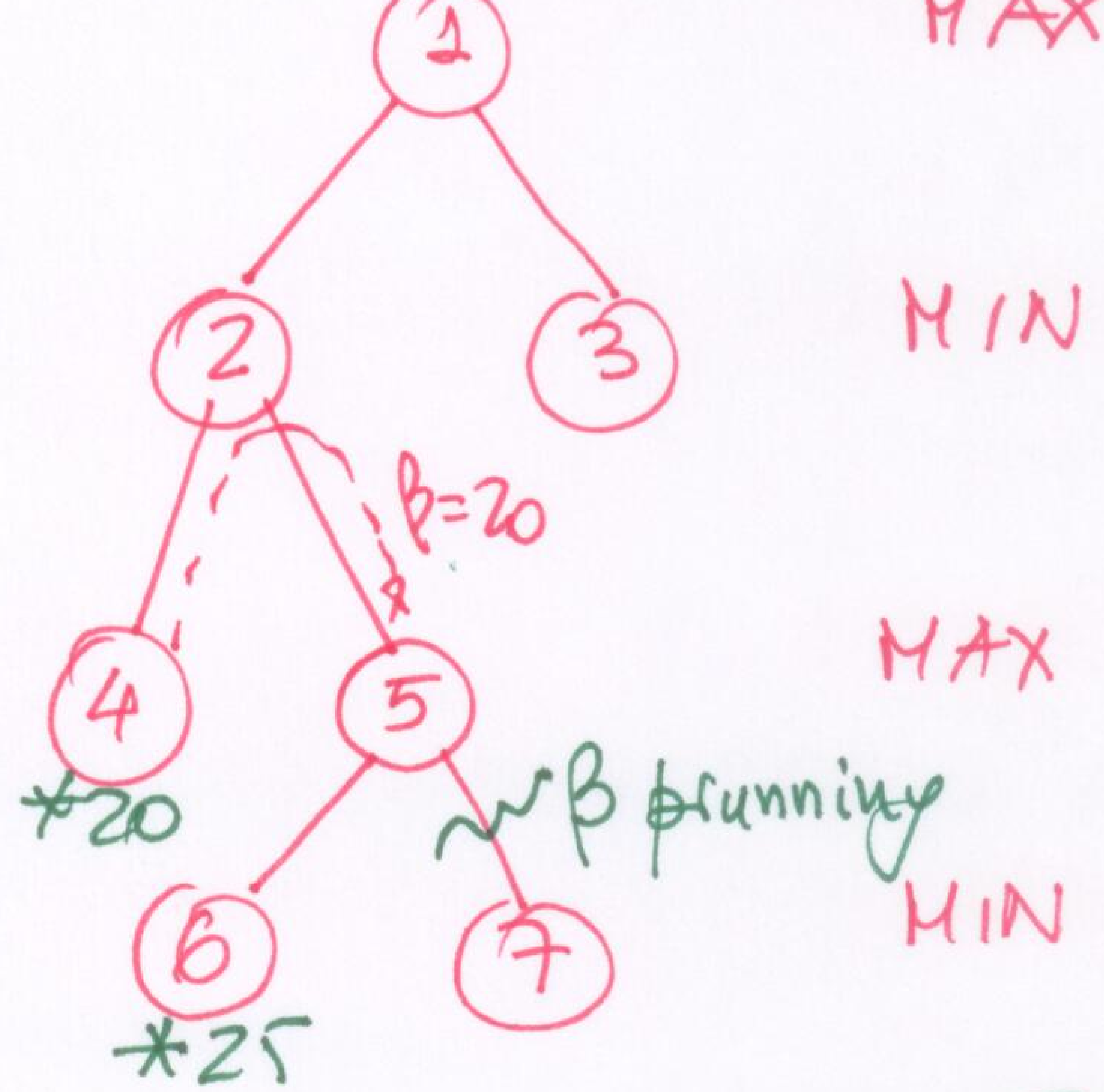
for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MIN(v, MAX-VALUE(s, α, β))

if $v \leq \alpha$ **then return** v

$\beta \leftarrow$ MIN(β, v)

return v



State	α	β	v	Function
1	$-\infty$	$+\infty$	$-\infty$	MAX-Value
2	$-\infty$	$+\infty$	$+\infty$	MIN-Value
4	return		20	MAX-Value
2	$-\infty$	20	20	MIN-Value
5	$-\infty$	20	$-\infty$	MAX-Value
6	return		25	MIN-Value
5	$-\infty$	20	25	MAX-Value

β pruning of the remaining successors.

State	α	β	v	Function
2	$-\infty$	20	20	MIN-value
1	20	$+\infty$	20	MAX-value

. . . .

function **ALPHABETA-SEARCH** (*state*)

returns *an action*

Inputs: *state,* current state in game

$v \leftarrow$ MAX-VALUE(*state*, $-\infty$, $+\infty$)

return the *action in SUCCESSORS*[*state*] with value *v*

---

function MAX-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow -\infty$

for *s* **in** SUCCESSORS[*state*] **do**

$v \leftarrow$ MAX(*v*, MIN-VALUE(*s*, α, β))

if $v \geq \beta$ **then return** *v*

$\alpha \leftarrow$ MAX (α, *v*)

return *v*

---

function MIN-VALUE (*state*, α, β) **returns** *a utility value*

Inputs: *state,* current state in game

α, the best value for MAX along the path to *state*

β, the best value for MIN along the path to *state*

if TERMINAL-TEST(*state*) **then return** UTILITY(*state*)

$v \leftarrow +\infty$

for *s* **in** SUCCESSORS[*state*] **do**

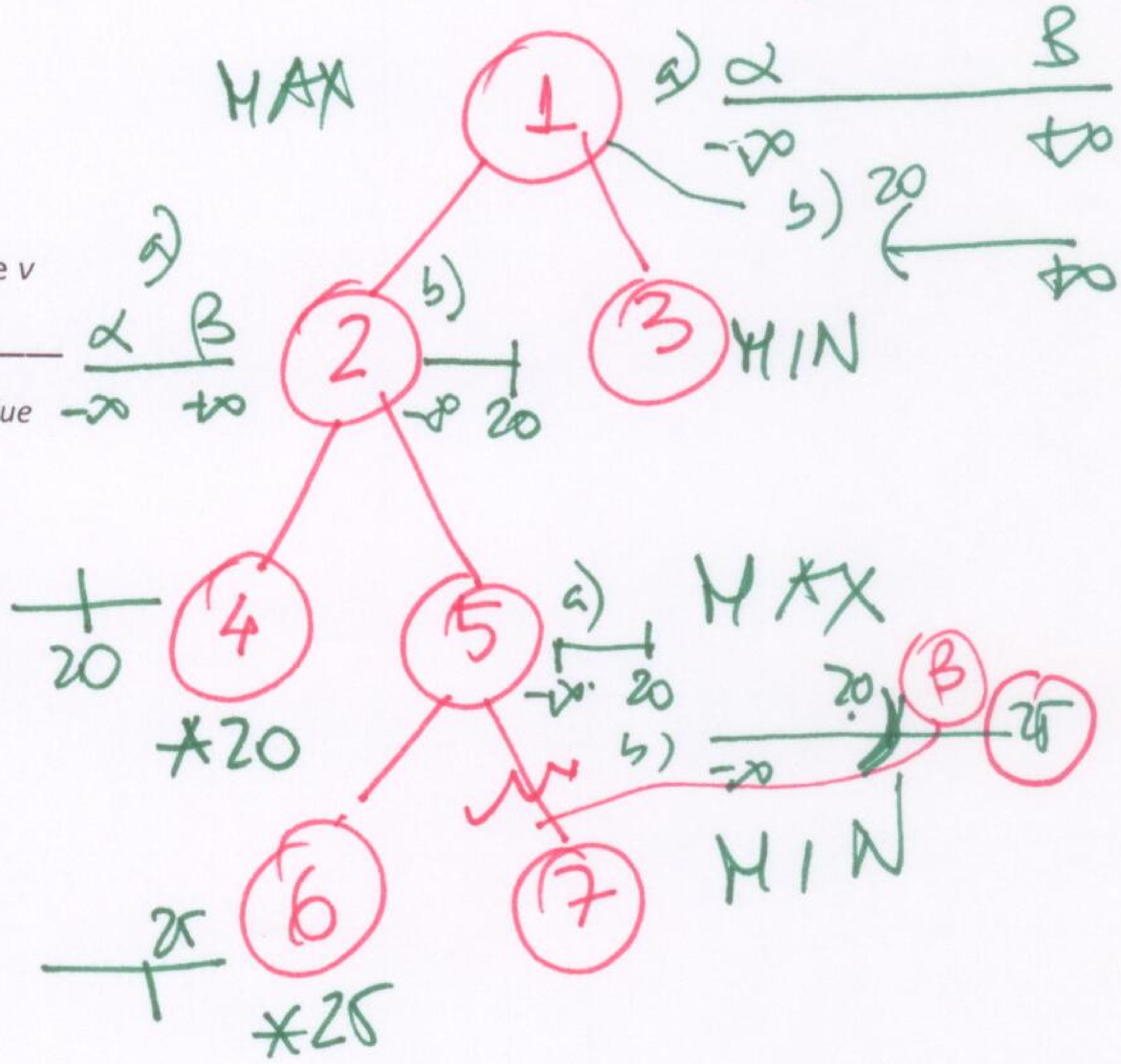
$v \leftarrow$ MIN(*v*, MAX-VALUE(*s*, α, β))

if $v \leq \alpha$ **then return** *v*

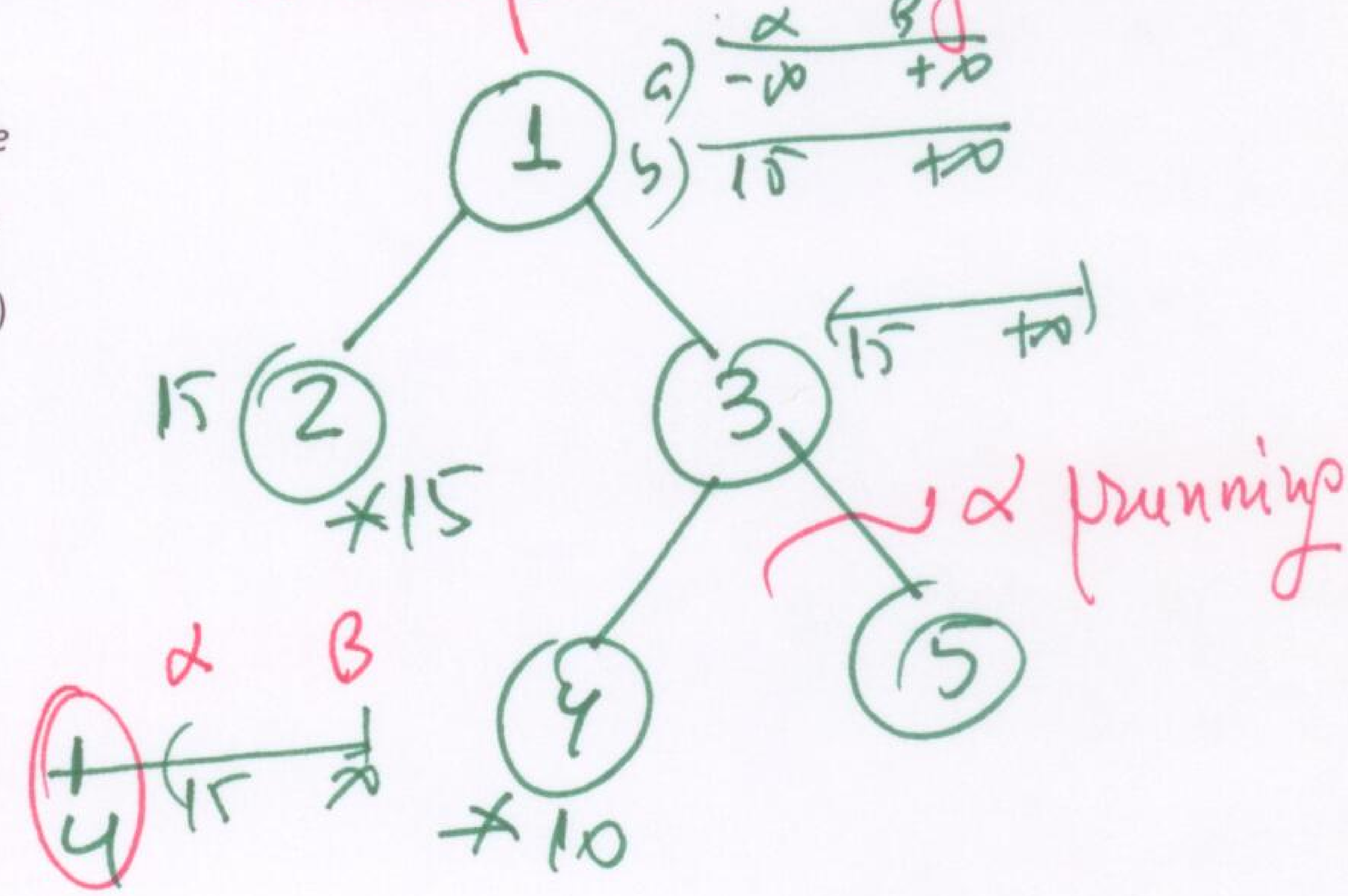
$\beta \leftarrow$ MIN(β, *v*)

return v

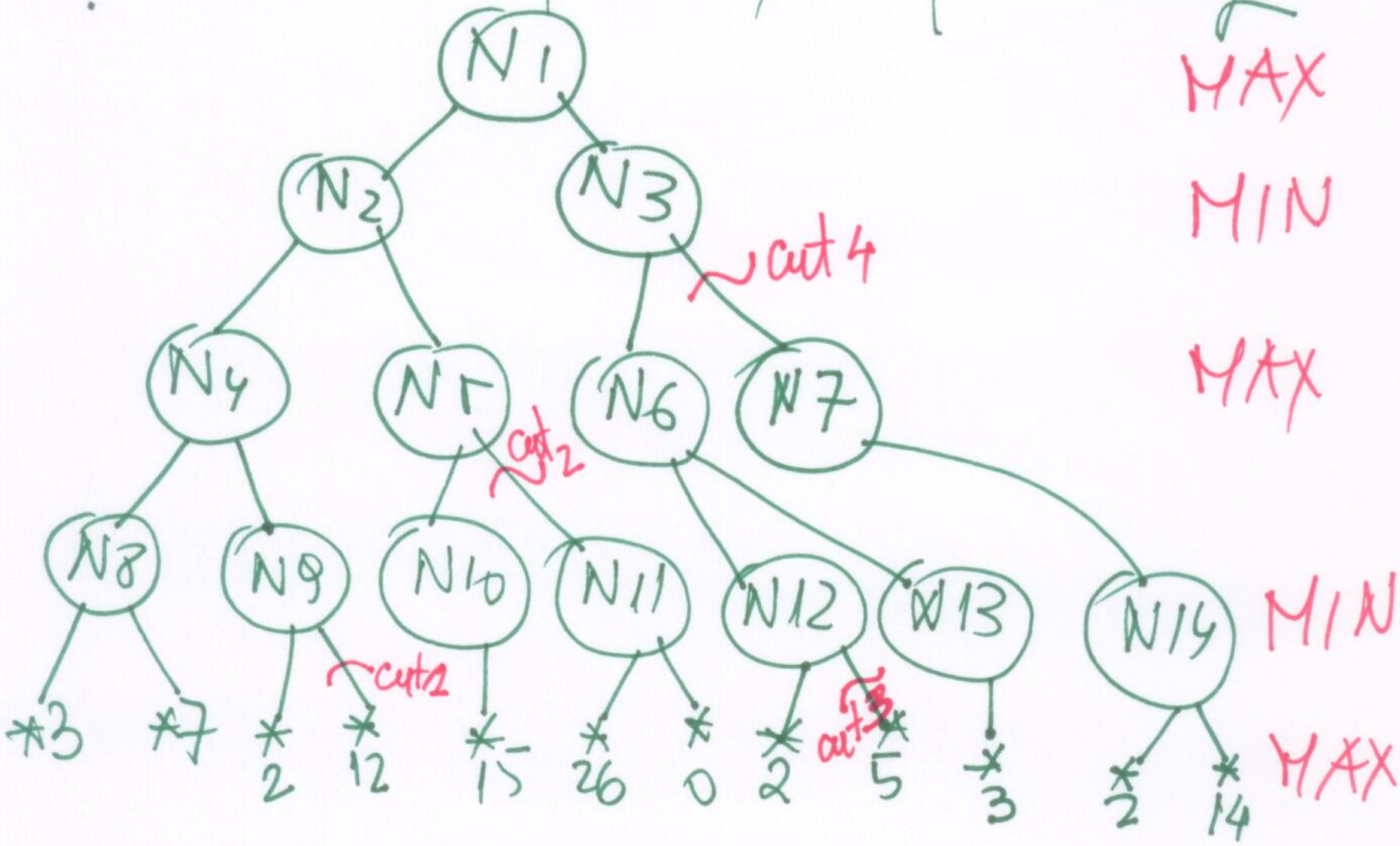
Example #2 again



Example #1 again



Extra-Example of α-β Prunning



Node	α	β	value
N 1	$-\infty$	$+\infty$	max-value = $-\infty$
N 2	$-\infty$	$+\infty$	min-value = $+\infty$
N 4	$-\infty$	$+\infty$	max-value = $+\infty$
N 8	$-\infty$	$+\infty$	min-value = $+\infty$
3	return 3		
N 8	$-\infty$	$\beta = 3$	min-value = 3
7	return 7		
N 8	$-\infty$	$\beta = 3$	min-value = 3
N 4	3	$+\infty$	max-value = 3
N 9	3	$+\infty$	min-value = $+\infty$

Node	α	β	value
2	return 2		
N9	3	$+\infty$	min-value = 2 < α = 3 α-cut 1
N4	3	$+\infty$	max-value = 3
N2	$-\infty$	β = 3	min-value = 3
N5	$-\infty$	3	max-value = $-\infty$
N10	$-\infty$	3	min-value = $+\infty$
15	return 15		
N10	$-\infty$	3	min-value = 15
N5	$-\infty$	3	max-value = 15 max-value $\geq \beta$ cut 2 (β)
N2	$-\infty$	3	min-value = 3
N1	3	$+\infty$	max-value = 3
N3	3	$+\infty$	min-value = $+\infty$
N6	3	$+\infty$	max-value = $-\infty$
N12	3	$+\infty$	min-value = $+\infty$
2	return 2		
N12	3	$+\infty$	min-value = 2 min-value = 2 < α = 3 α cut#3

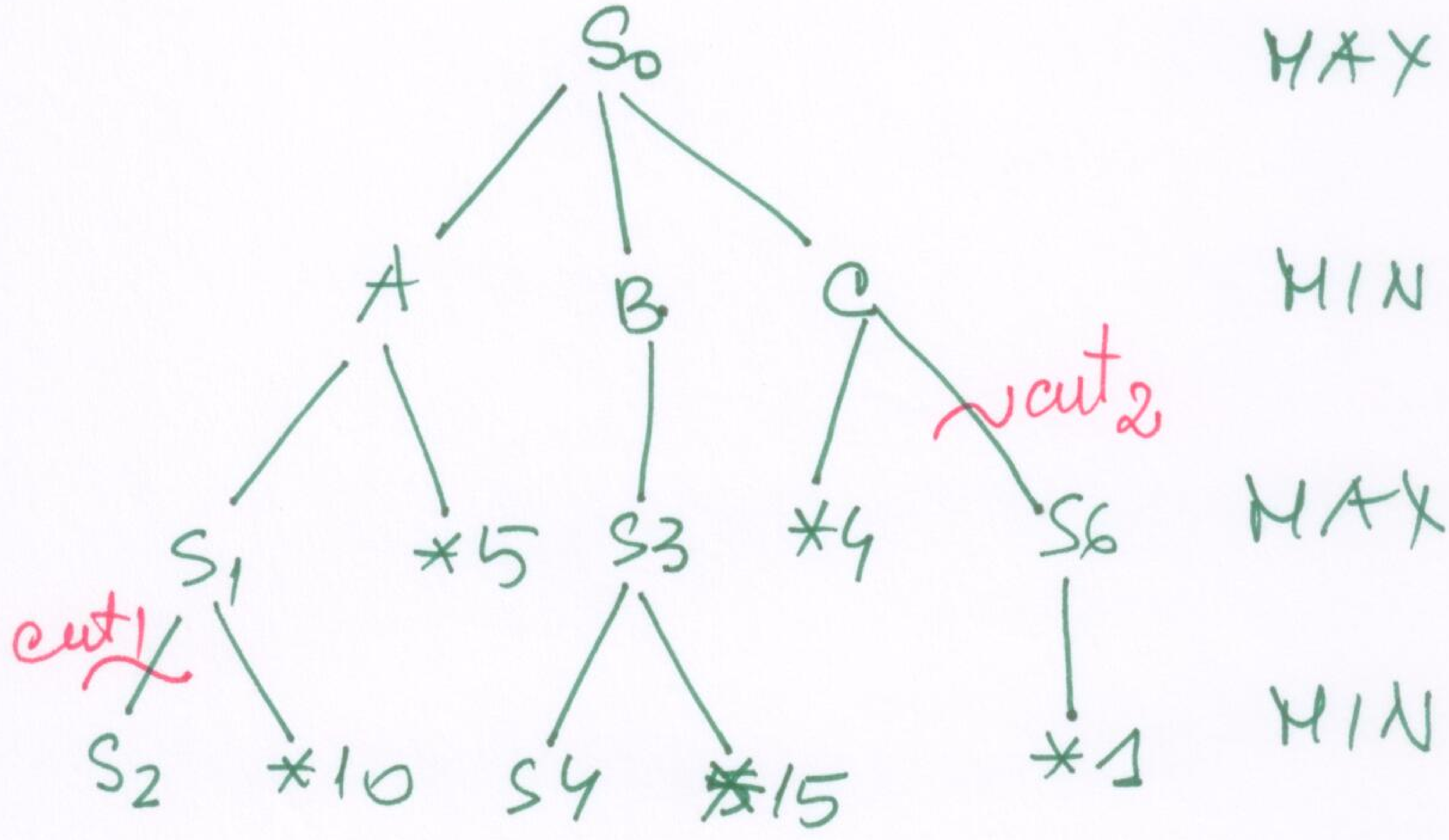
Node	α	β	value
N6	3	$+\infty$	max-value = 2
N13	3	$+\infty$	min-value = $+\infty$
3	return 3		
N13	3	3	min-value = 3 No successors!
N6	3	$+\infty$	max-value = 3 no cut
N3	3	$+\infty$	min-value = 3

min-value = 3 $\leq \alpha = 3$

$\Downarrow$

α-cut #4

Is α-β prunning cutting any nodes of this game tree?



Node	α	β	value
S0	$-\infty$	$+\infty$	max-value = $-\infty$
A	$-\infty$	$+\infty$	min-value = $+\infty$
~~S1~~	~~$-\infty$~~	~~$+\infty$~~	~~max-value = $-\infty$~~ or better → not yet!
5	return 5		
A	$-\infty$	5	min-value = 5
S1	$-\infty$	5	max-value = $-\infty$
10	return 10		
S1	$-\infty$	5	max-value = 10 β

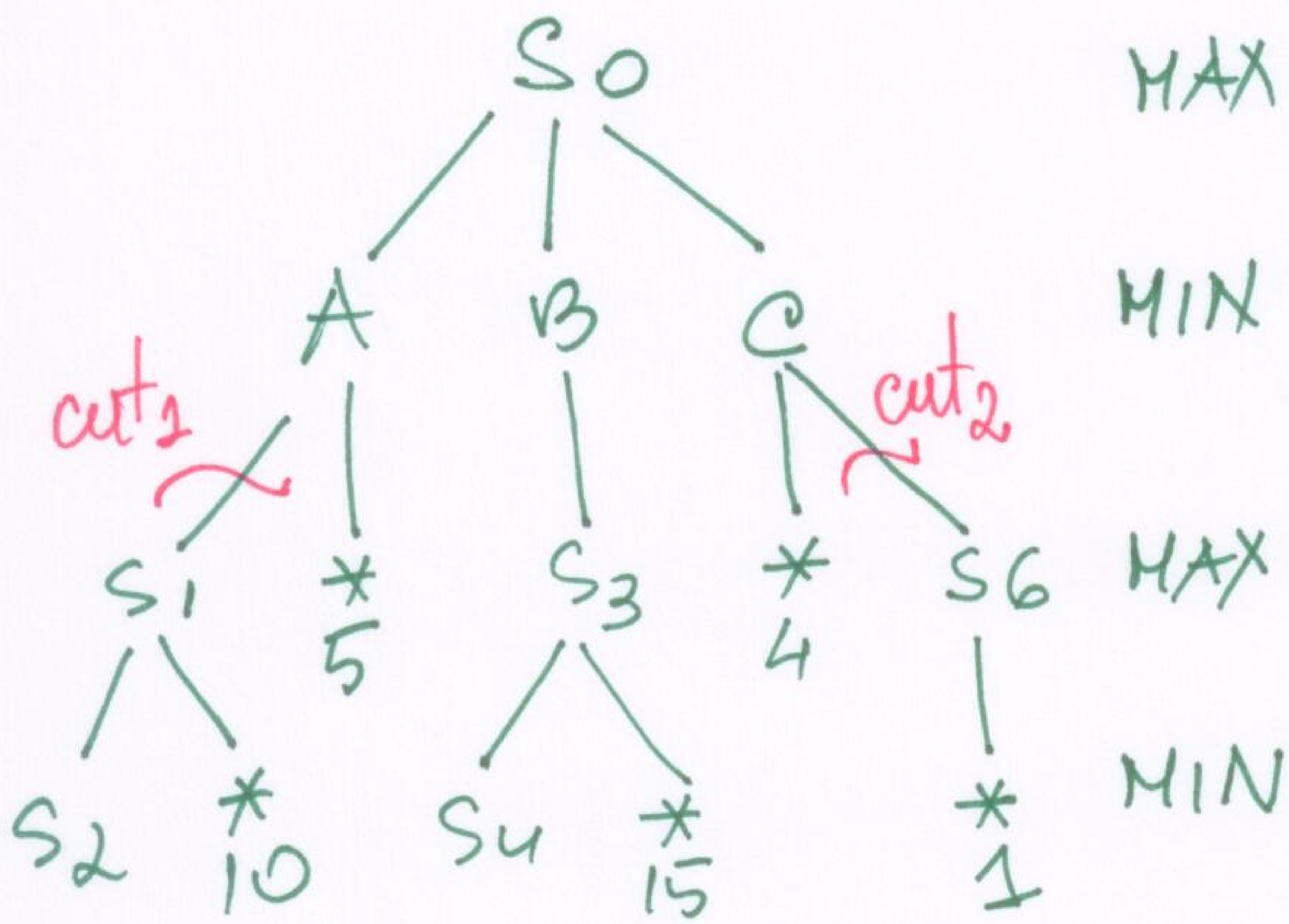
max-value = 10 $\geq \beta$ = 5 cut1

Node	α	β	value
A	$-\infty$	5	min-value = 5
S0	5	$+\infty$	max-value = 5
B	5	$+\infty$	min-value = $+\infty$
S3	5	$+\infty$	max-value = $-\infty$
15	return	15	
S3	15	$+\infty$	max-value = 15
B	5	~~$+\infty$~~ 15	We do not visit S4 min-value = 15
S0	15	$+\infty$	max-value = 15
C	15	$+\infty$	min-value = $+\infty$
4	return	4	
C	15	$+\infty$	min-value = 4 min-value = $4 \le \alpha = 15$ cut 2 → α-cut
S0	15	$+\infty$	max-value = 15

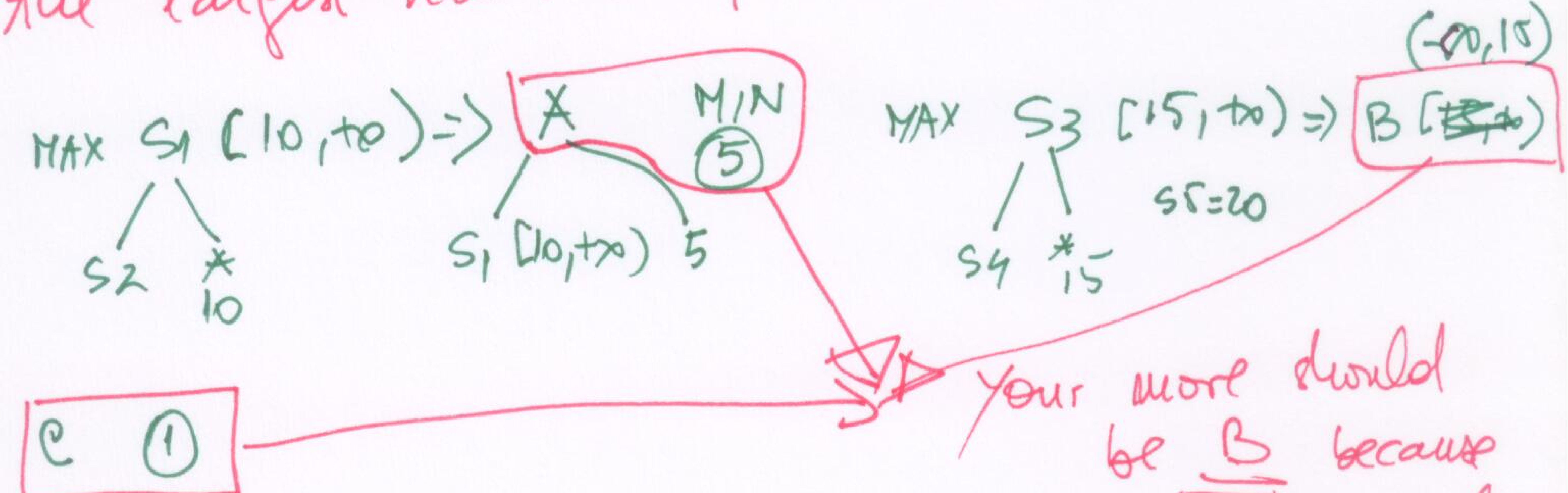
⇒ no other successors

so $\alpha = \beta =$ max-value $= 15$

Can you find the "killer move" on the same game tree?

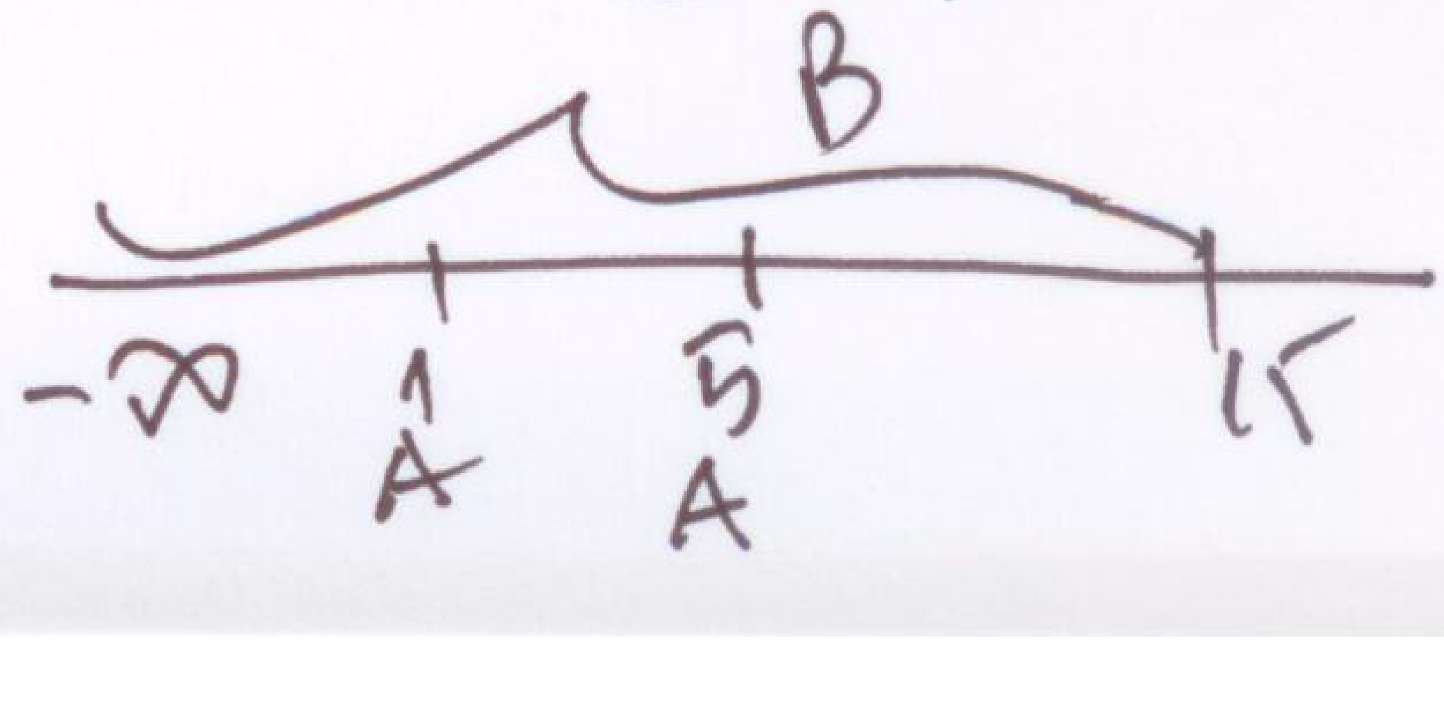


Answer: which move should MAX chose at the beginning (A?, B?, C?) such that we can prune the largest numbers of nodes?



Your move should be B because A is a MAX node.

let us see how the prunning operates:



Node	α	β	value
S0	$-\infty$	$+\infty$	max-value $= -\infty$
B	$-\infty$	$+\infty$	min-value $= +\infty$
S3	$-\infty$	$+\infty$	max-value $= -\infty$
15	return 15		
S3	15	$+\infty$	max-value $= 15$
B	$-\infty$	15	min-value $= 15$ — we do not visit S4!
S0	15	$+\infty$	max-value $= 15$
A	15	$+\infty$	min-value $= +\infty$ — where should we go next, A or C?
5	return 5		
A	15	$+\infty$	min-value $= 5$ — min-value $= 5 \leq \alpha = 15$ cut (α-cut) by removing the other successors of A
S_0	15	$+\infty$	max-value $= 15$
C	15	$+\infty$	min-value $= +\infty$
4	return 4		

Node	α	β	value
C	15	$+\infty$	min-value = 4

min-value = 4 $\leq \alpha = 15$

cut 2 (α-cut)

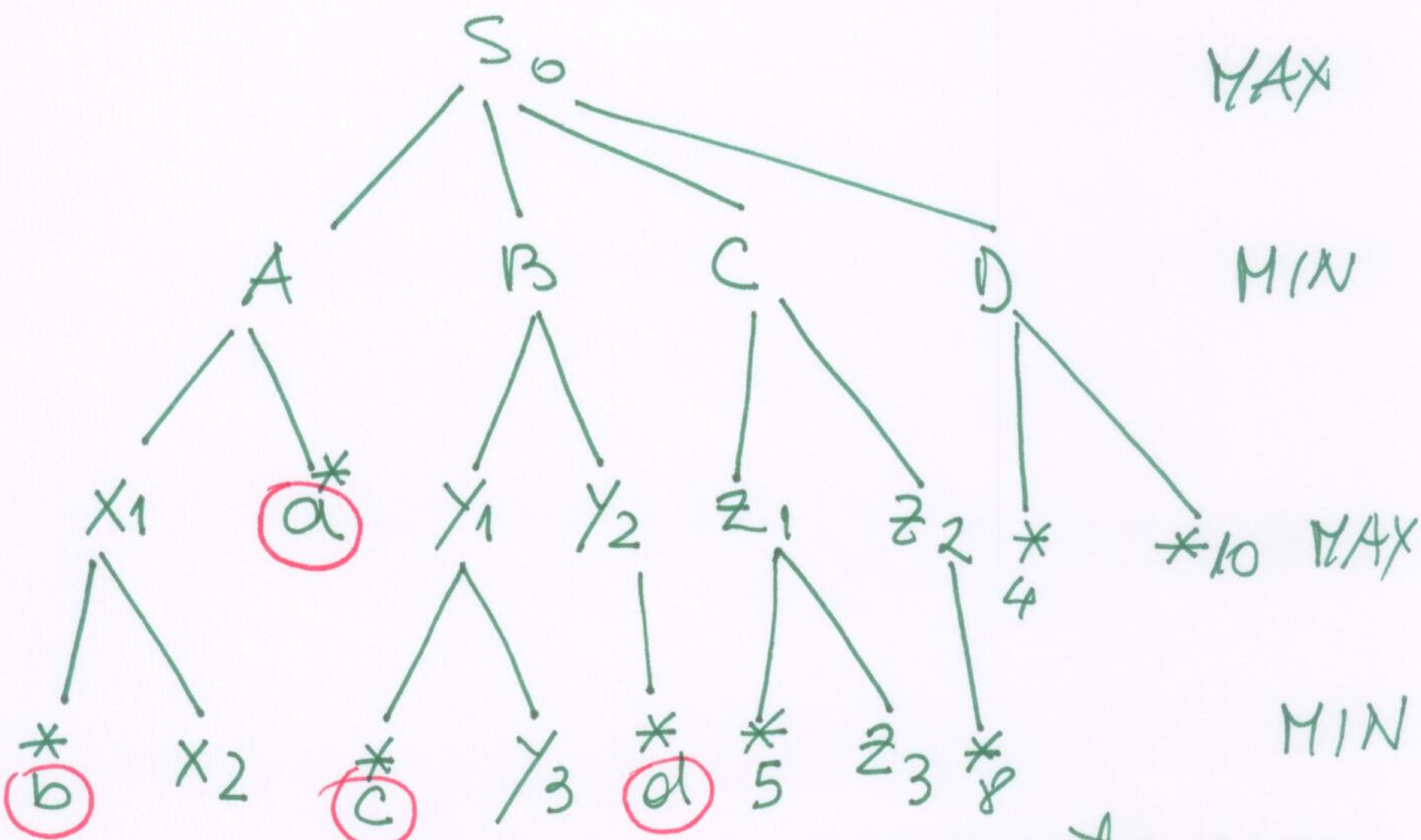
→ remove the remaining nodes of C (the nodes at S6)

Analysis

How many nodes were not inspected now? S1, S2, *10, S6, *1 } 5 nodes
1 2 3 4 5

How many nodes we did not inspect before? S2, S6, *1 } 3 nodes
1 2 3

Consider the following game tree:



a) How is α-β prunning operating on this tree?

b) Find the values of a and b that will determine a cut → show the cut on the tree.

c) Find the values of c and d that will determine another cut → show the cut on the tree

let us find all answers at once!

Node	α	β	value
S_0	$-\infty$	$+\infty$	$-\infty$ = maxvalue
A	$-\infty$	$+\infty$	$+\infty$ = min value
a	return a		
A	$-\infty$	a	min value = a
X_1	$-\infty$	a	max value = $-\infty$
b	return b		
X1	$-\infty$	a	max value = b

We can have a cut$_1$ if $\boxed{b \geq a}$

β-cut

X_2 no longer visited

A	$-\infty$	a	min-value = a
S_0	~~$-\infty$~~ a	$+\infty$	max-value = a
B	~~$-\infty$~~ a	$+\infty$	min-value = $+\infty$
4	return 4		
D	~~$-\infty$~~ a	$+\infty$	min-value = 4

We can have a cut$_2$ if $\boxed{a \geq 4}$

α-cut

Now we have $\dfrac{\text{we cut 10}}{b \geq a \quad 4}$

Node	α	β	value
S_0	~~$-\infty$~~	$+\infty$	max-value = a
C	a	$+\infty$	min-value = $+\infty$
z_2	a	$+\infty$	max-value = $-\infty$
8	return 8		
z_2	a	$+\infty$	max-value = 8
C	a	~~8~~ $+\infty$	min-value = 8

if $8 \le a$ we have a cut 3

is it possible?

we know $a \in [4, +\infty)$

if $a \in [4, 8]$ YES

$\Downarrow$

$b \in (8, +\infty)$

α-cut

cut 71

Node	α	β	value
S_0	a	$+\infty$	max-value = a
B	a	$+\infty$	min-value = $+\infty$
Y 2	a	$+\infty$	max-value = ~~a~~ $-\infty$
return d			
Y 2	a or d	$+\infty$	max-value = d

Node	α	β	value
Y2	d	$+\infty$	max-value $= d$

$$\frac{d \quad a}{d} \quad \begin{matrix} \text{if } a \geq d & \underline{\text{case 1}} \\ \text{if } d \geq a & \text{case 2} \end{matrix}$$

<u>Case 1</u>: $a \in [4, 8] \Rightarrow d \in (-\infty, 4)$

<u>Case 2</u> $a \in [4, 8] \Rightarrow d \in (8, +\infty)$

We follow <u>case 1</u> first:

Node	α	β	value
Y2	a	$+\infty$	max-value $= d$
B	a	$+\infty$	min-value $= d$

d = min-value $\leq \alpha = a$

$\Downarrow$

α cut $\Rightarrow$ do not visit node Y1 any <u>more</u>

$c \in (-\infty, +\infty)$

S0	a	$+\infty$	max-value $= a$

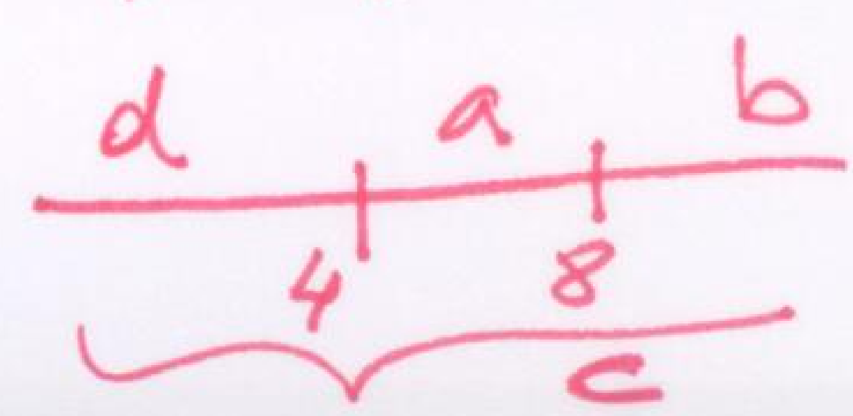
no more successors

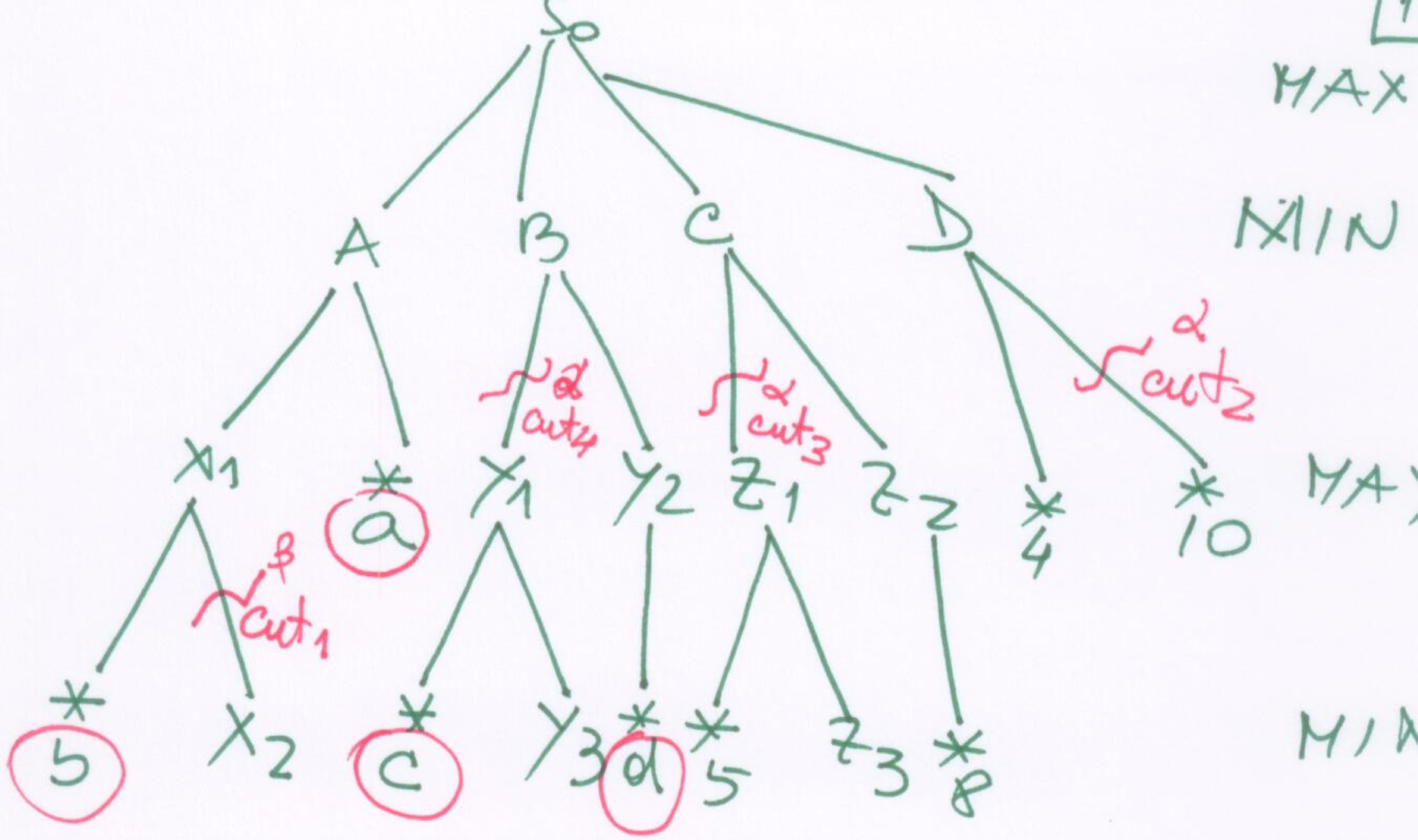
$\Rightarrow \underline{\beta = a}$

Solution 1 : $a \in [4, 8]$, $b \in (8, +\infty)$

$c \in (-\infty, +\infty)$

$d \in (-\infty, 4)$





---

let us also consider case 2:

$a \in [4, 8]$; $d \in (8, +\infty)$; $b \in (8, +\infty)$

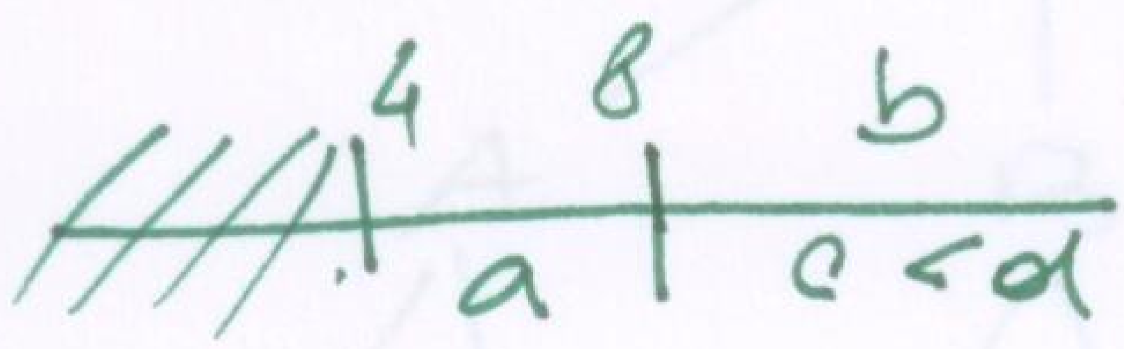
Node	α	β	value
Y2	d	$+\infty$	max-value = d
B	a	$+\infty$	min-value = d d > a ⇒ v > α ok, no cut!
Y1	a	$+\infty$	max-value = $-\infty$
C	return	c	
Y1	a	$+\infty$	max-value = c if c ≥ a } case 2.1 max-value = a otherwise } case 2.2.

Case 2.1. $a \in [4,8]$ $d \in (8,+\infty)$
$c \in (8,+\infty)$ ~~d~~ $b \in (8,+\infty)$

Node	α	β	value
Y1	c	$+\infty$	c = max-value
B	a	$+\infty$	min-value = c if $c<d$ (case 2.1.1); min-value = d if $d<c$ (case 2.1.2)
S_0	---	no other cuts !	

we have solutions for a, b, c and d as:

Case 2.1.1 : $a \in [4,8]$ $b \in (8,+\infty)$
$c, d \in (8,+\infty)$ but $c<d$



2.1.2 $a \in [4,8]$ $b \in (8,+\infty)$
$c, d \in (8,+\infty)$ but $d<c$