JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Adversarial Search

601.464
Artificial Intelligence
TR 10.30AM—11.45PM

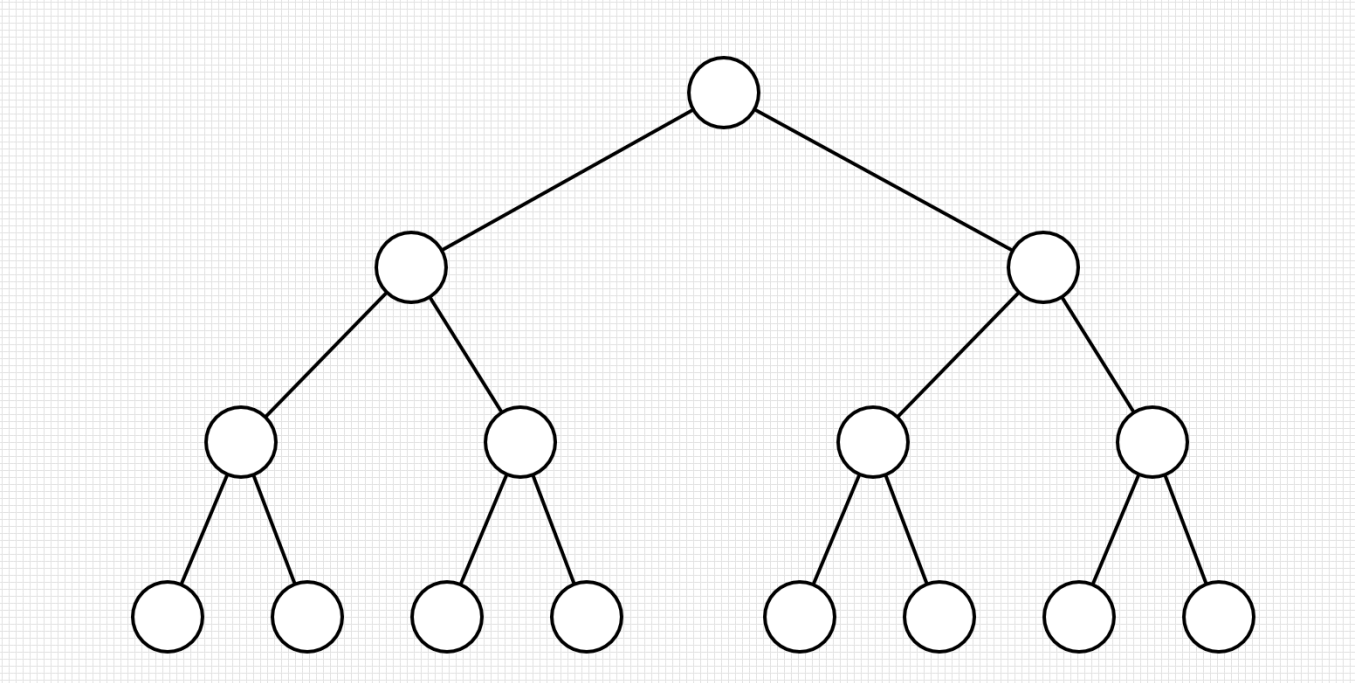
Material

- Vocabulary
- Game play (chess)
- MINIMAX Algorithm
- MINIMAX Algorithm + $\alpha\beta$

Vocabulary

- Branching factor (b) = 2
  - Number of possible moves available at each node
- Depth (d) = 3
- Terminal nodes
  - At d = 0, terminal nodes = 1
  - At d = 1, terminal nodes = 2
  - At d = 2, terminal nodes = 4
  - At d = 3, terminal nodes = 8

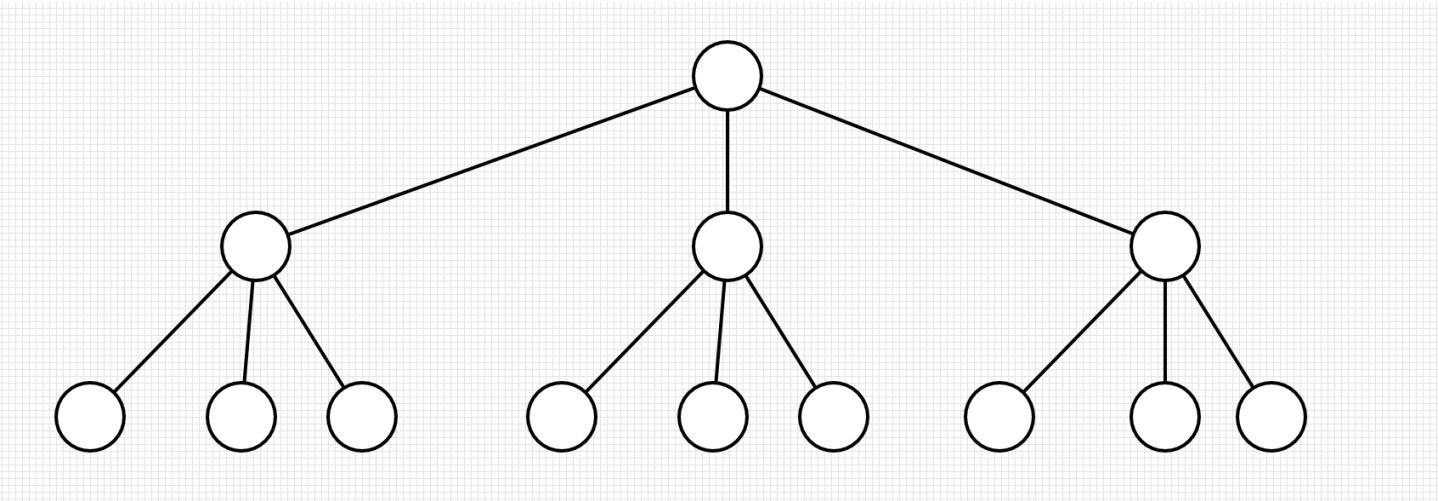
  …
  - At d, terminal nodes = b^d



Vocabulary

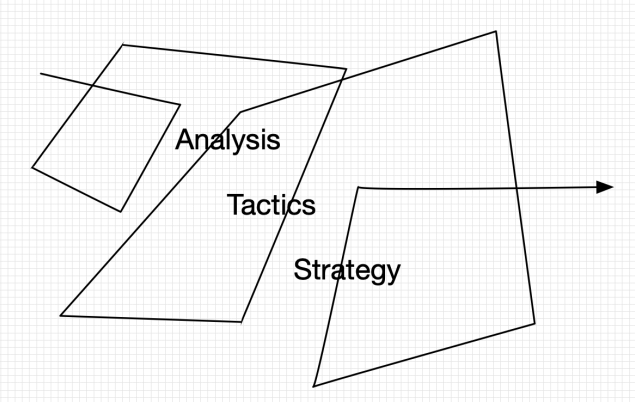
- Branching factor (b) = 3
  - Number of possible moves available at each node
- Depth (d) = 2
- Terminal nodes
  - At d = 0, terminal nodes = 1
  - At d = 1, terminal nodes = 3
  - At d = 2, terminal nodes = 9

  …
  - At d, terminal nodes = b^d



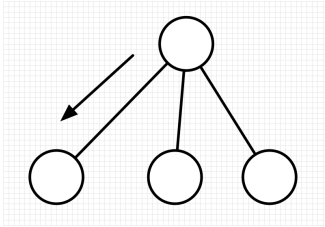
Chess

1 Human

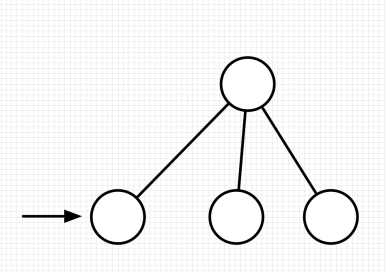


2 IF-THEN RULES

- If a move is available, then take that action



3 Look ahead to next depth and evaluate

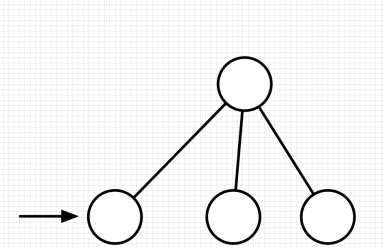


4 British museum

- Expand the tree all the way to the end and evaluate the terminal nodes to pick the best move

Chess

3 Look ahead to next depth and evaluate



But how do we evaluate a board, i.e., assign a number or utility or payoff or reward to it?

Perform *static* evaluation

"Static" since opponent reaction not modeled after the point (depth) at which the evaluation is performed

Usual choice is a *linear combination* of features you care about

- Number of pieces (f_1, for instance)
- Pawn structure (f_2, for instance)
- Control of center
- King safety
- And many more! (I'm told)

$$S = \sum_{1}^{N} w_i * f_i \qquad \sum_{1}^{N} w_i = 1$$

$$S = (w_1 f_1) + \ldots + (w_N f_N)$$

Chess

Terminal nodes in chess turn out to be 10^{120}

So using British Museum would mean evaluating 10^{120} boards

$\pi.10^{7}.10^{9}$ nanoseconds in a year

10^{80} atoms in the universe number of computers

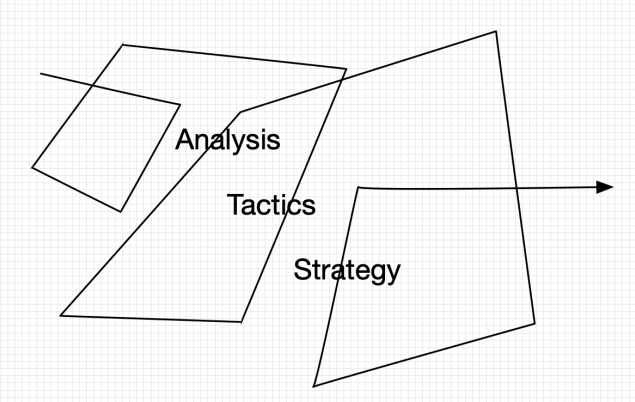
At nanosecond computations would mean 10^{22} years of completion time

4 British museum

- Expand the tree all the way to the end and evaluate the terminal nodes to pick the best move

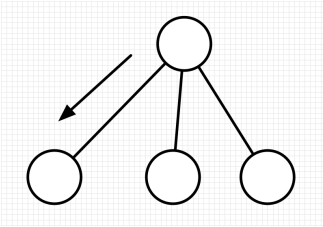
Chess

1 Human

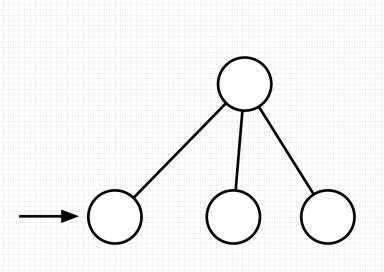


2 IF-THEN RULES

- If a move is available, then take that action



3 Look ahead to next depth and evaluate



4 British museum

- Expand the tree all the way to the end and evaluate the terminal nodes to pick the best move

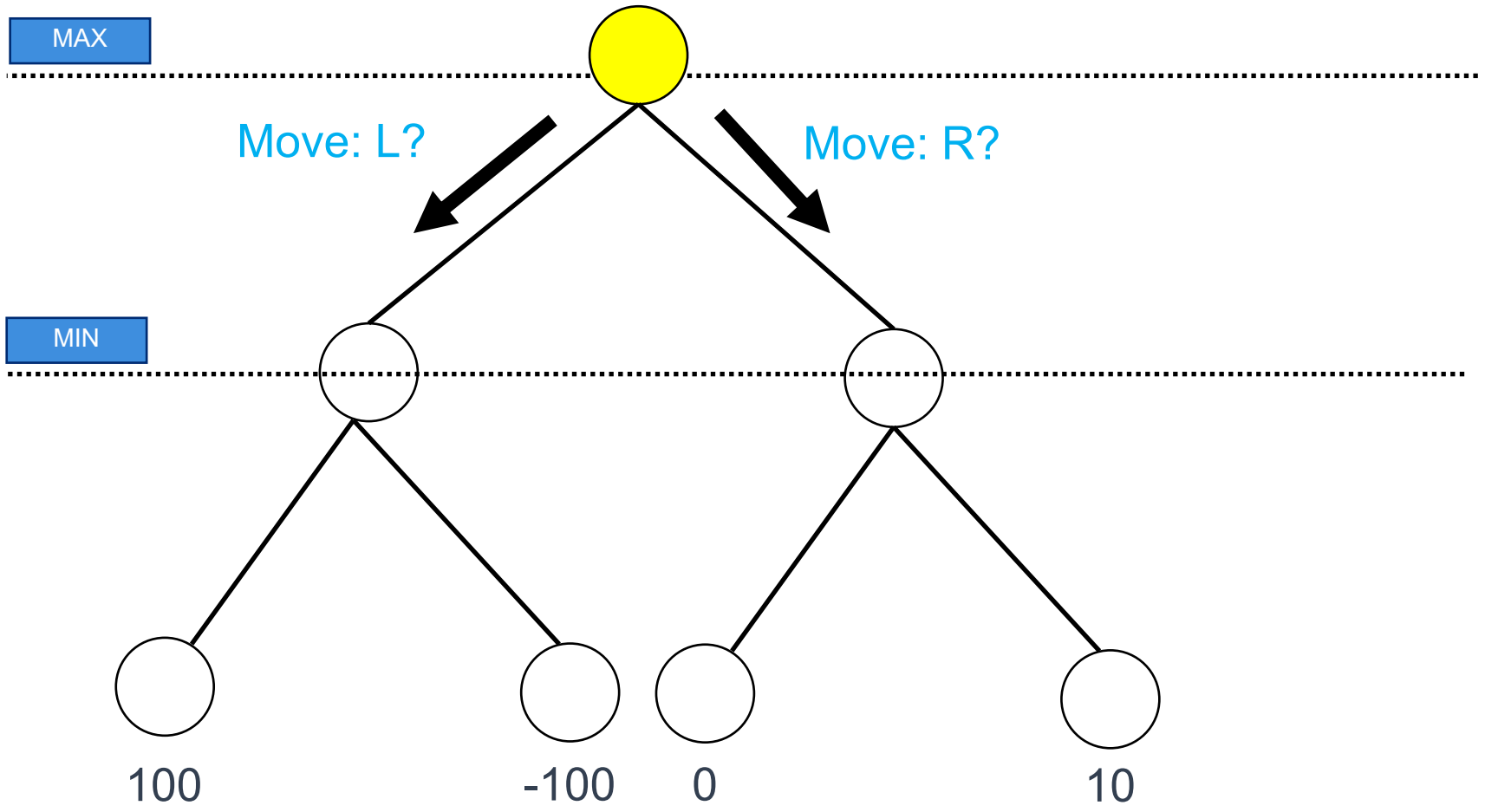
5 Look ahead as much as possible

- MINIMAX
- MINIMAX + $\alpha\beta$

MINIMAX

P1 needs to make a decision

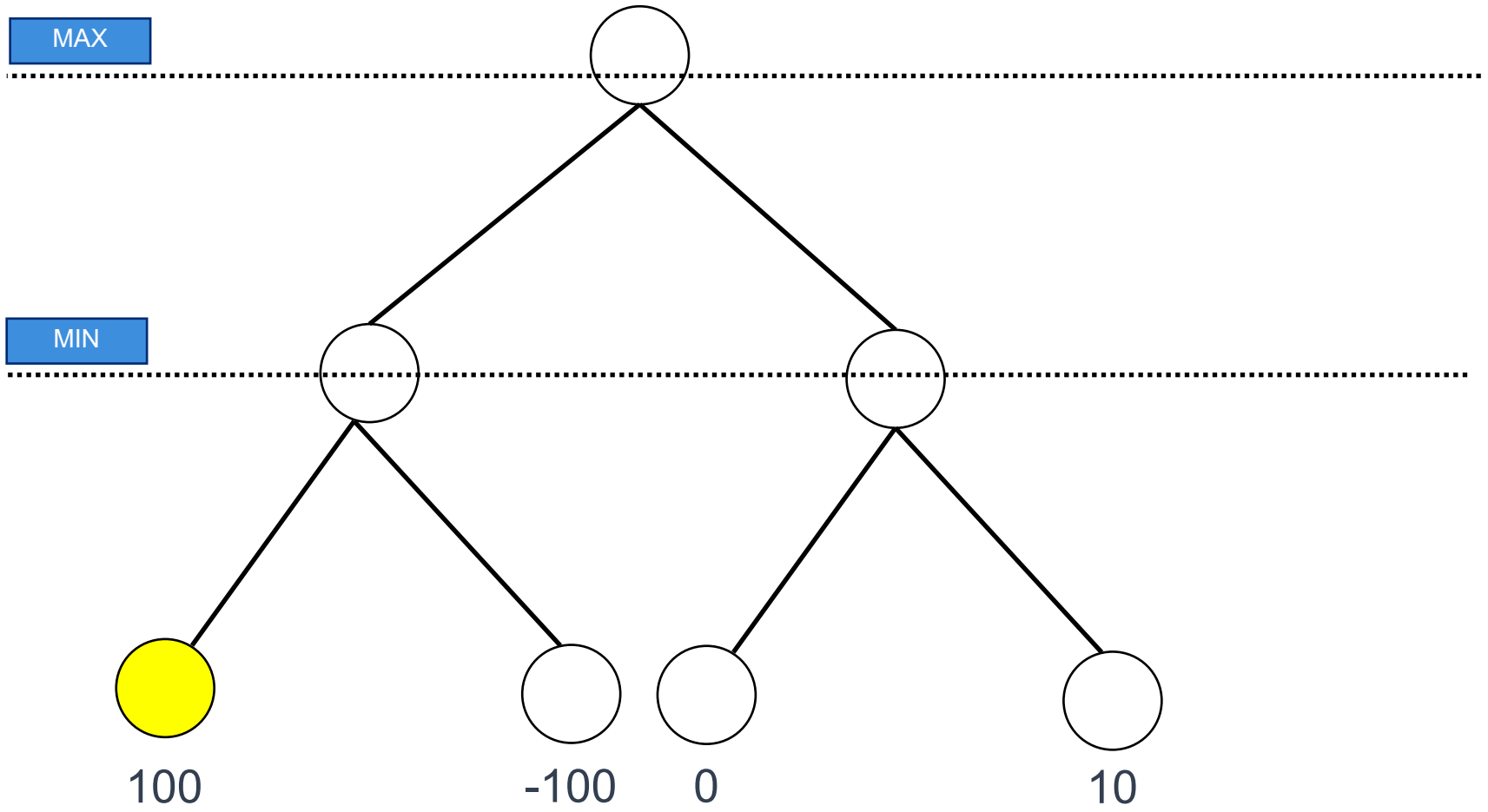
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Start at the left!

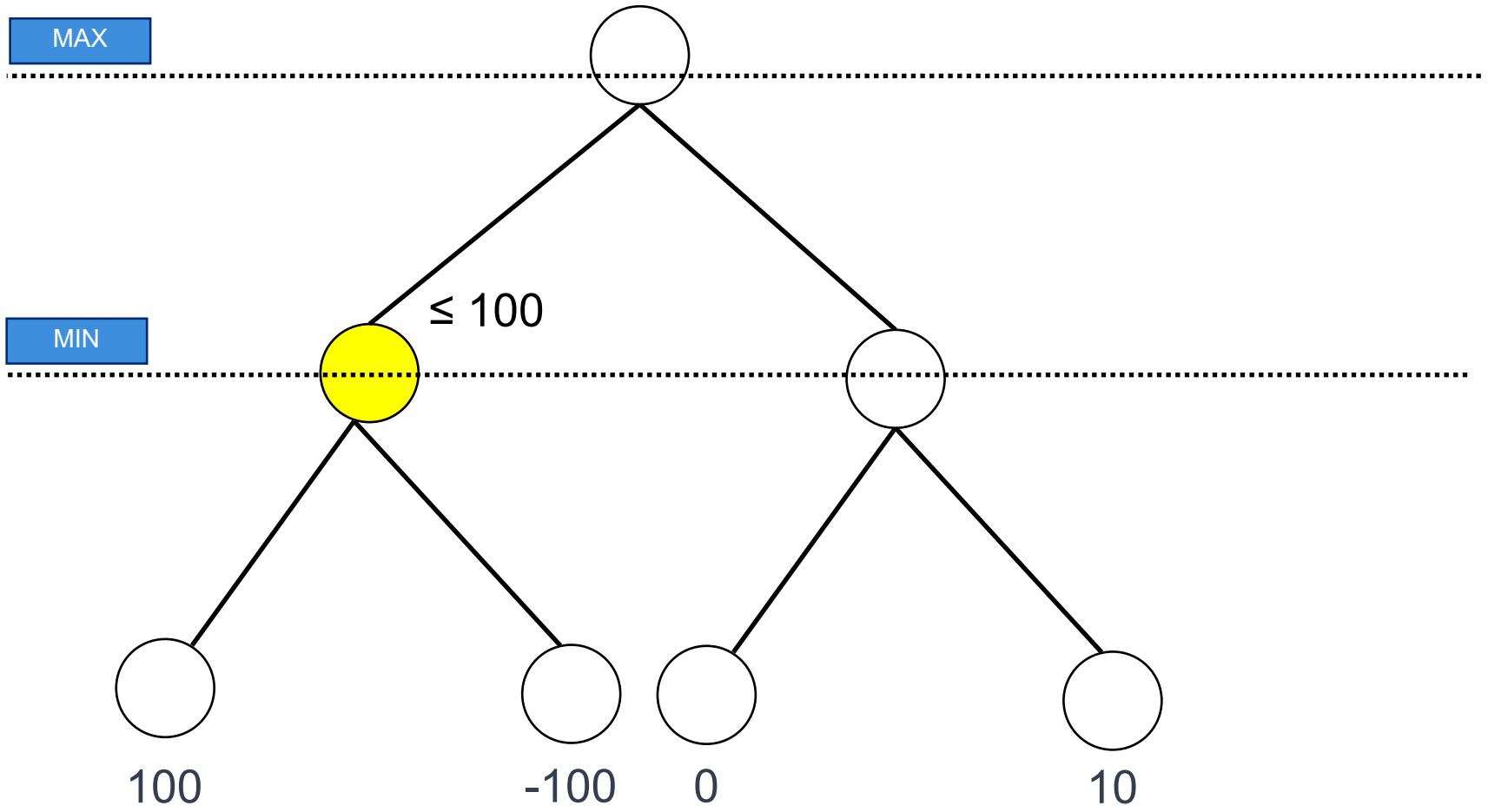
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Propagate up

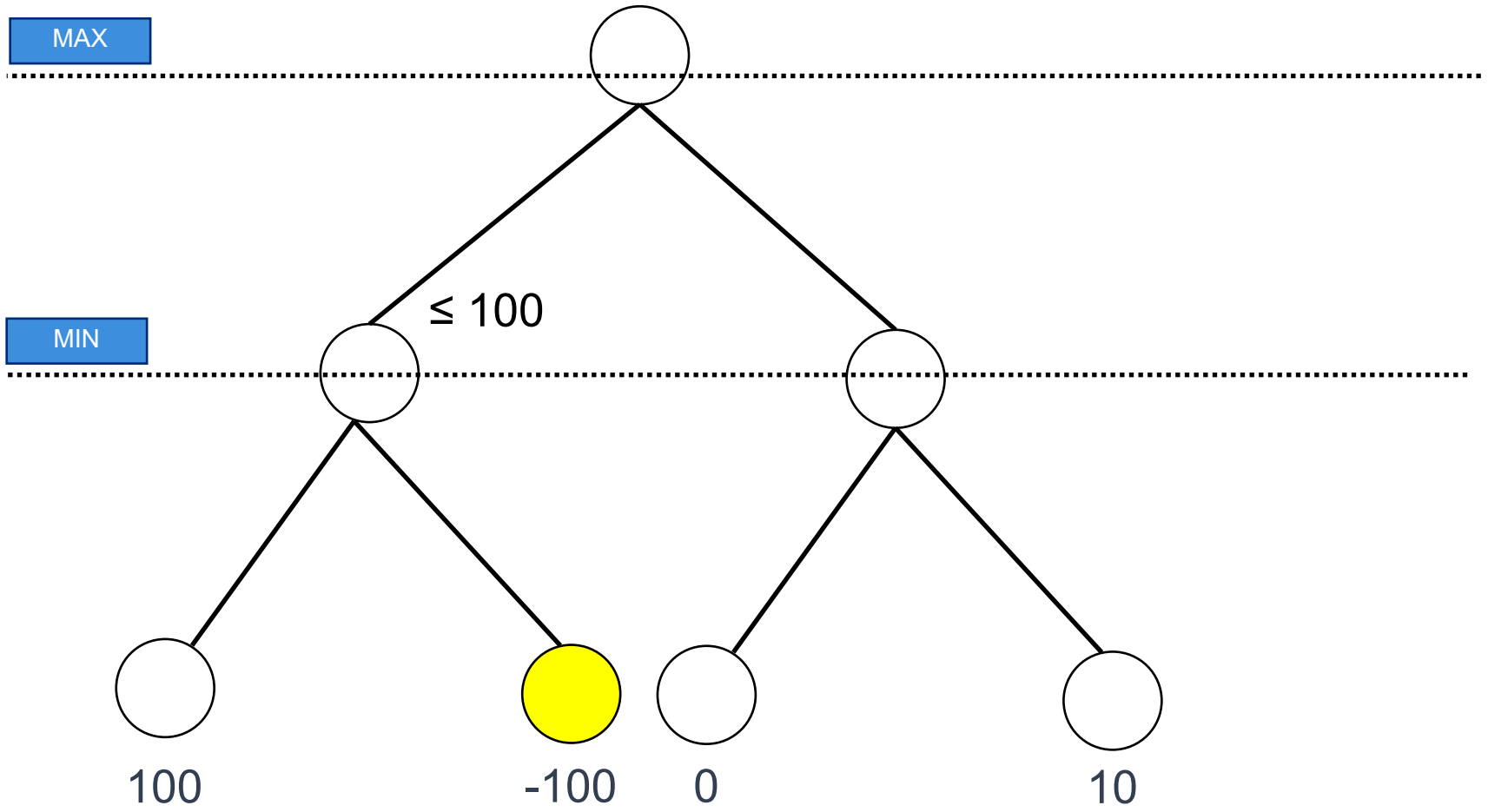
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Next terminal node

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)

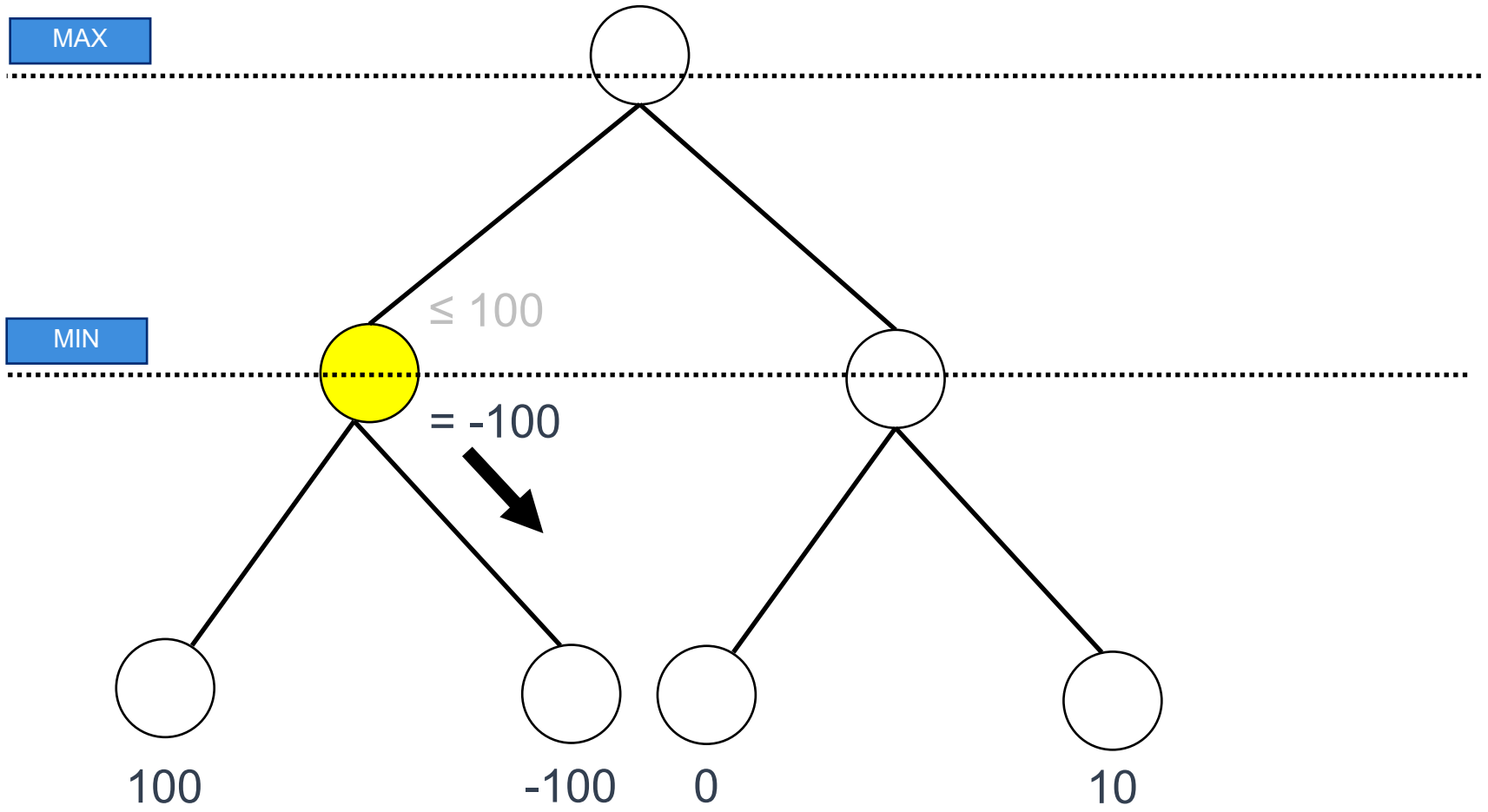


MINIMAX

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)

Propagate up

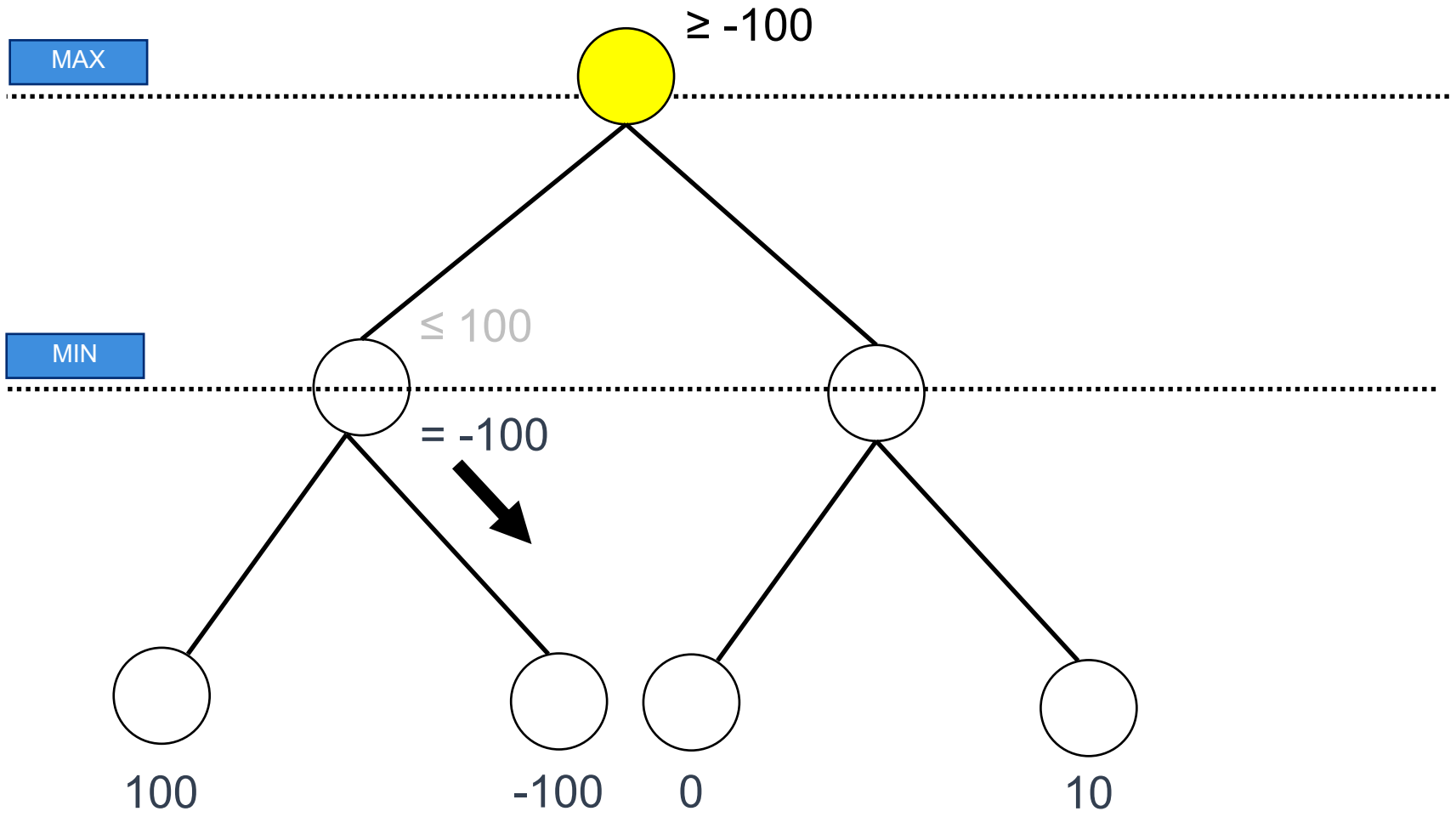
Update



MINIMAX

Propagate up

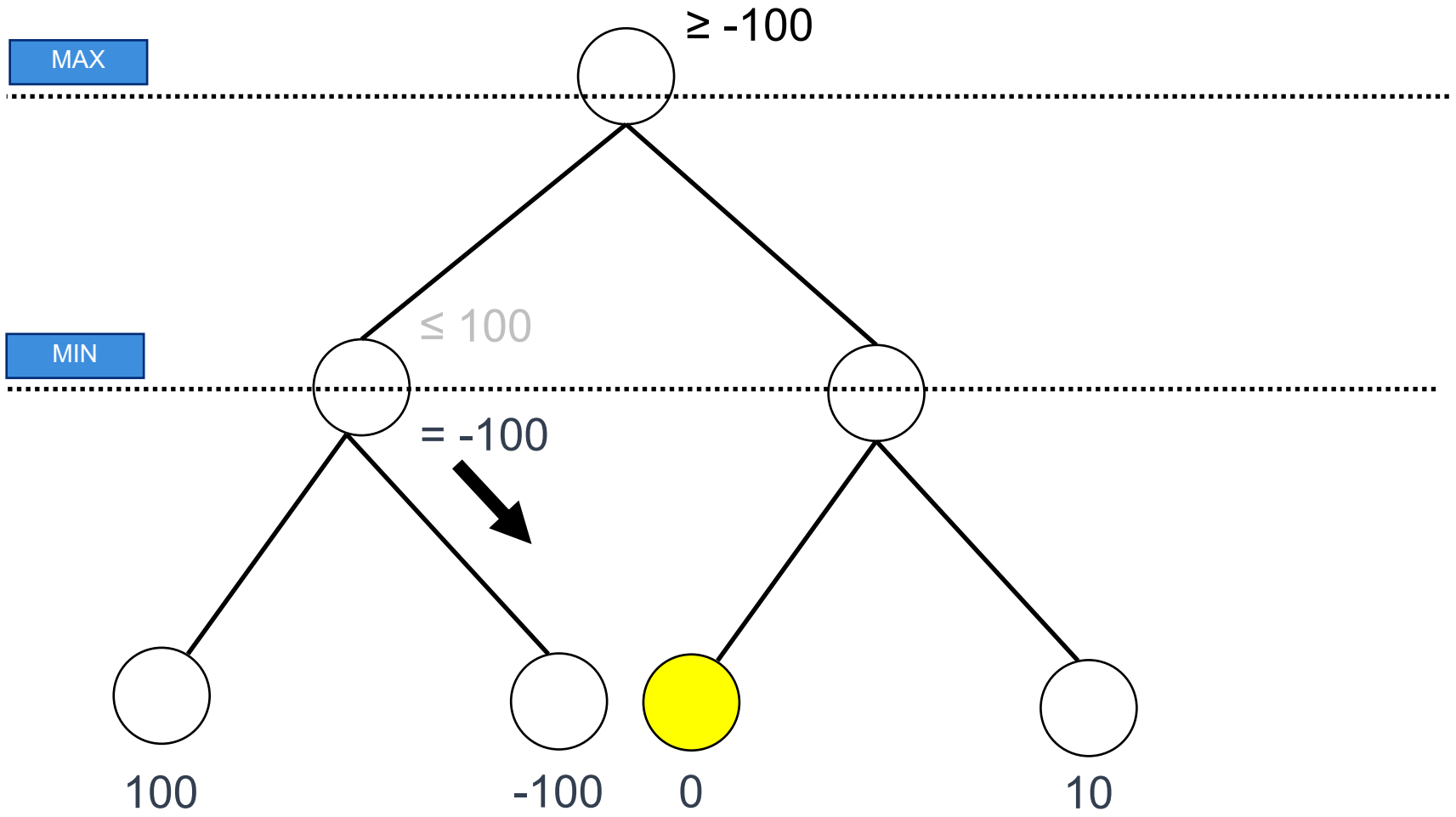
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Next terminal node

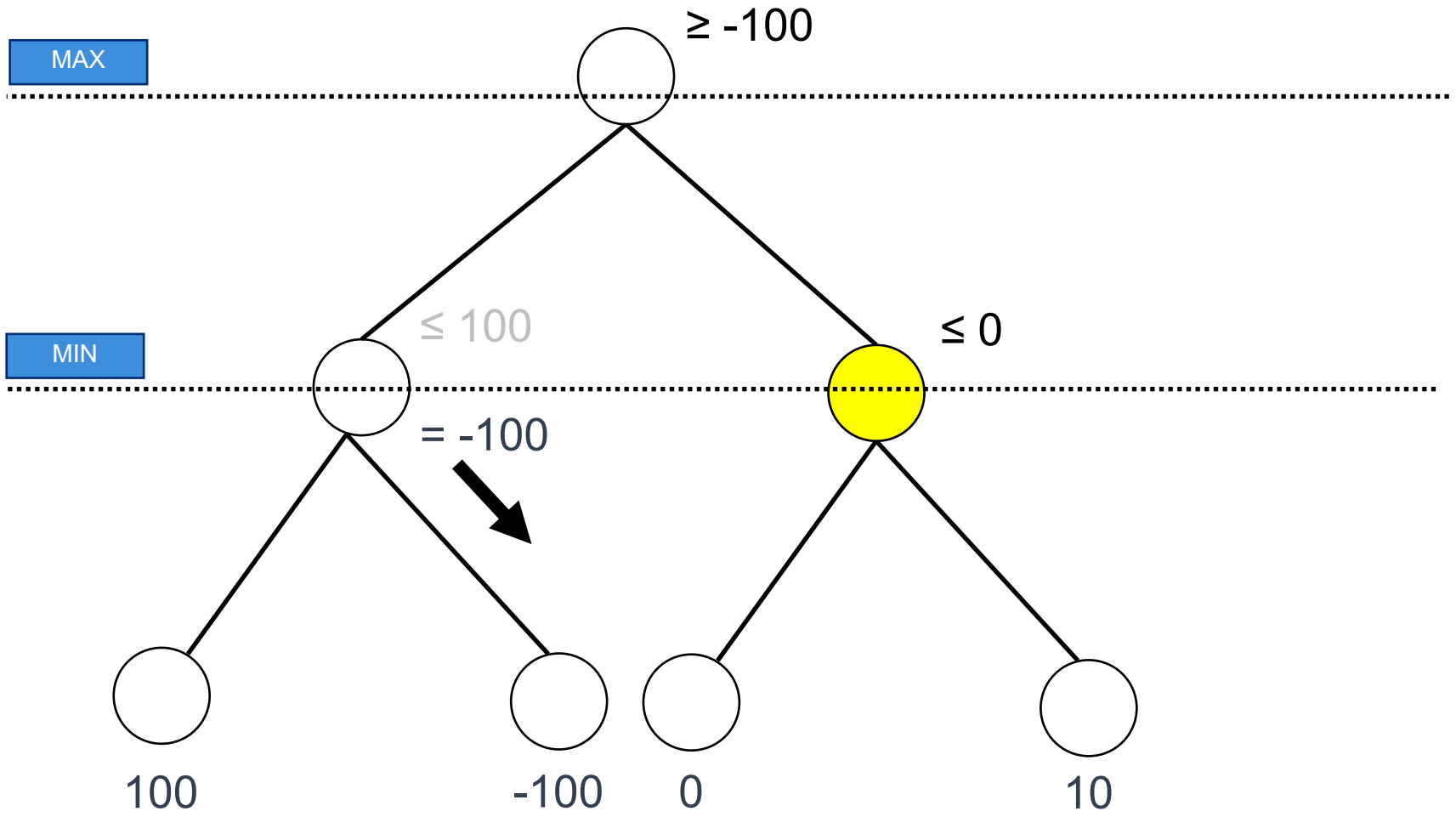
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Propagate up

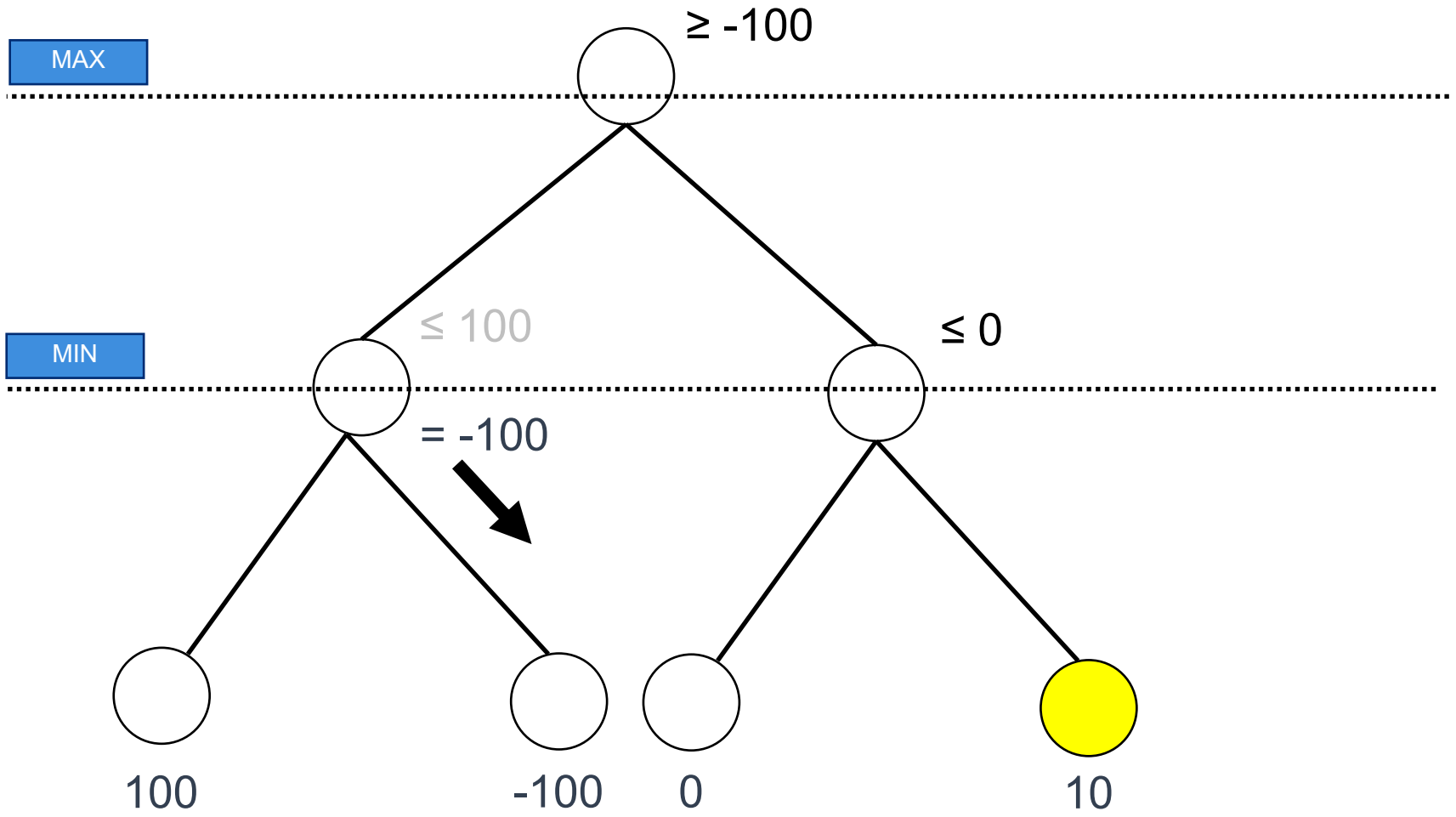
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



MINIMAX

Next terminal node

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)

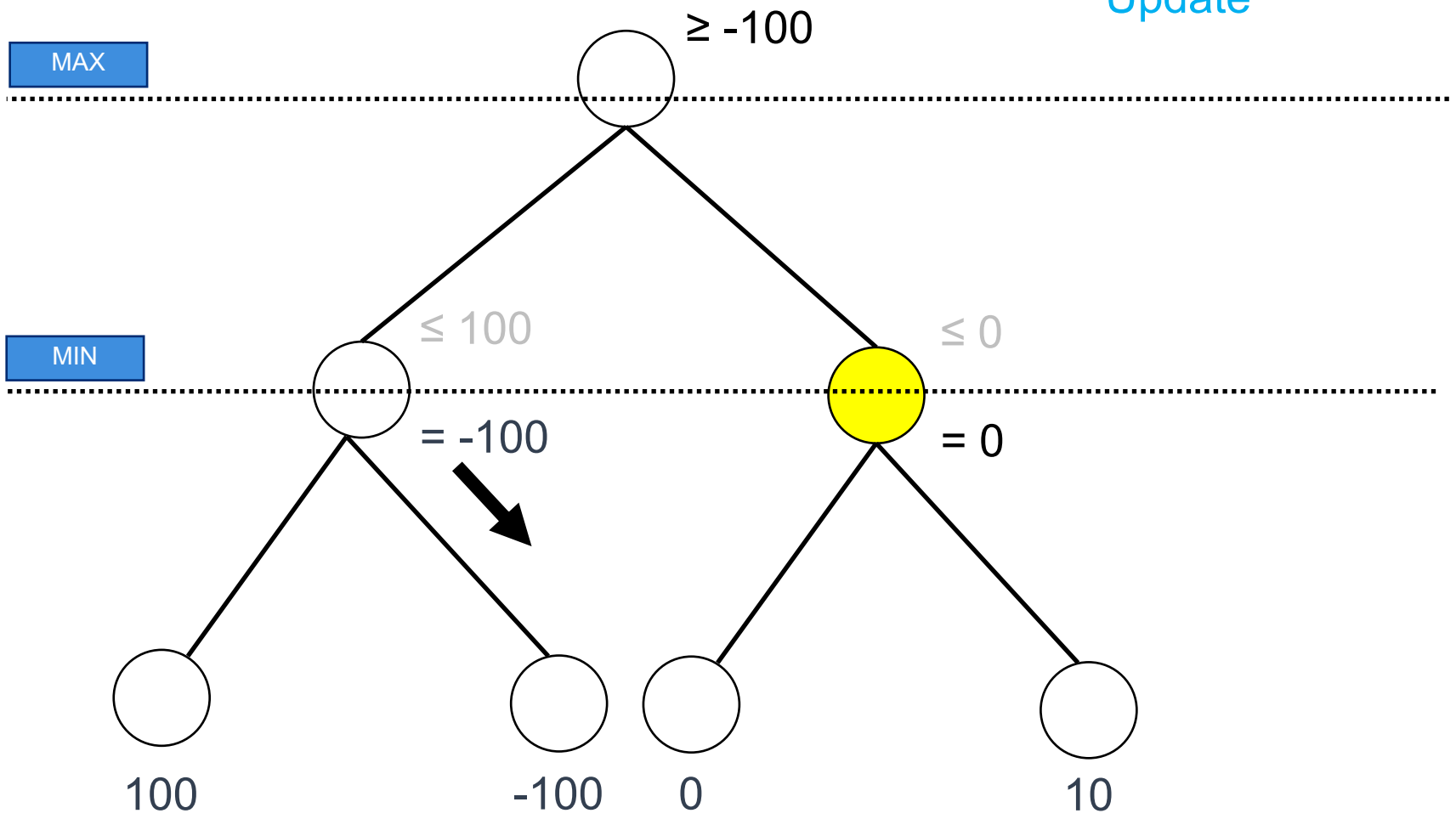


MINIMAX

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)

Propagate up

Update

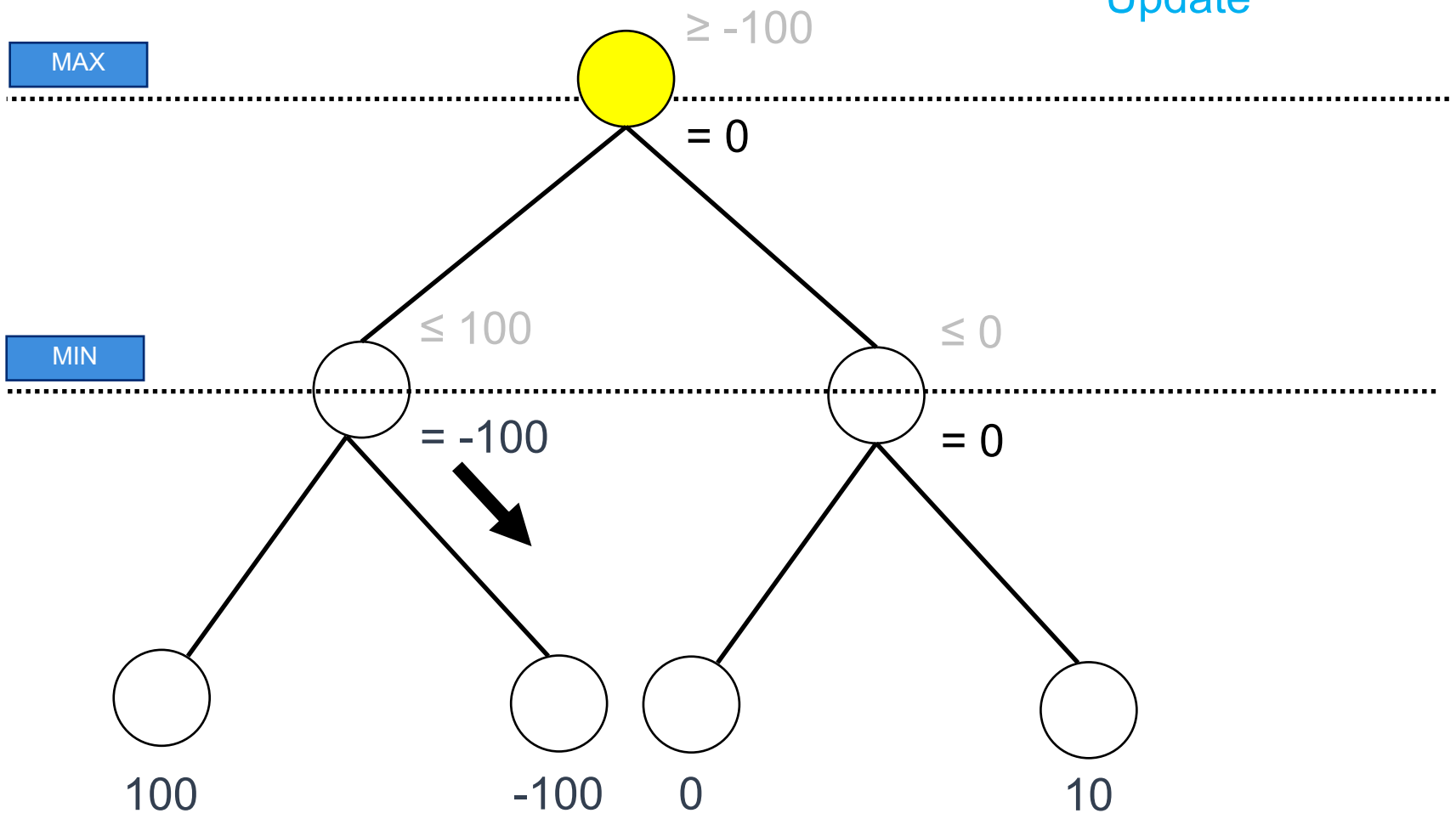


MINIMAX

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)

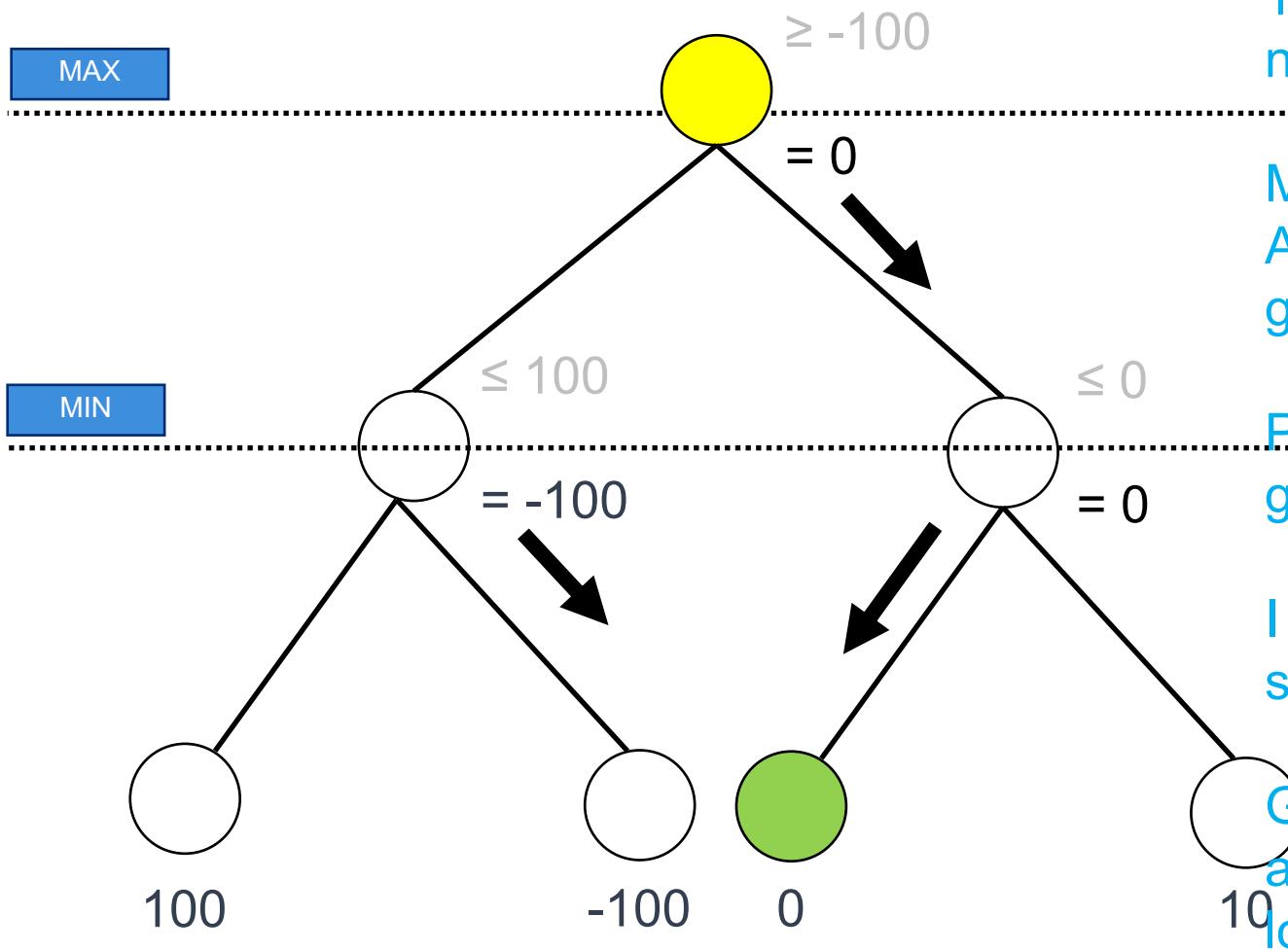
Propagate up

Update



MINIMAX

- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Run static evaluations on ***ALL*** the terminal nodes at the depth you care about
- Propagate the values back to the very top (the MAX node)



Move R!

The value of this node is 0

Makes sense. Adversarial game.

P2 won't let me get 100.

I won't let P2 serve me -100.

Game ends up in a compromise location.

MINIMAX

- b^d calculations needed before we could even start (getting all the board evaluations)

MINIMAX + $\alpha\beta$

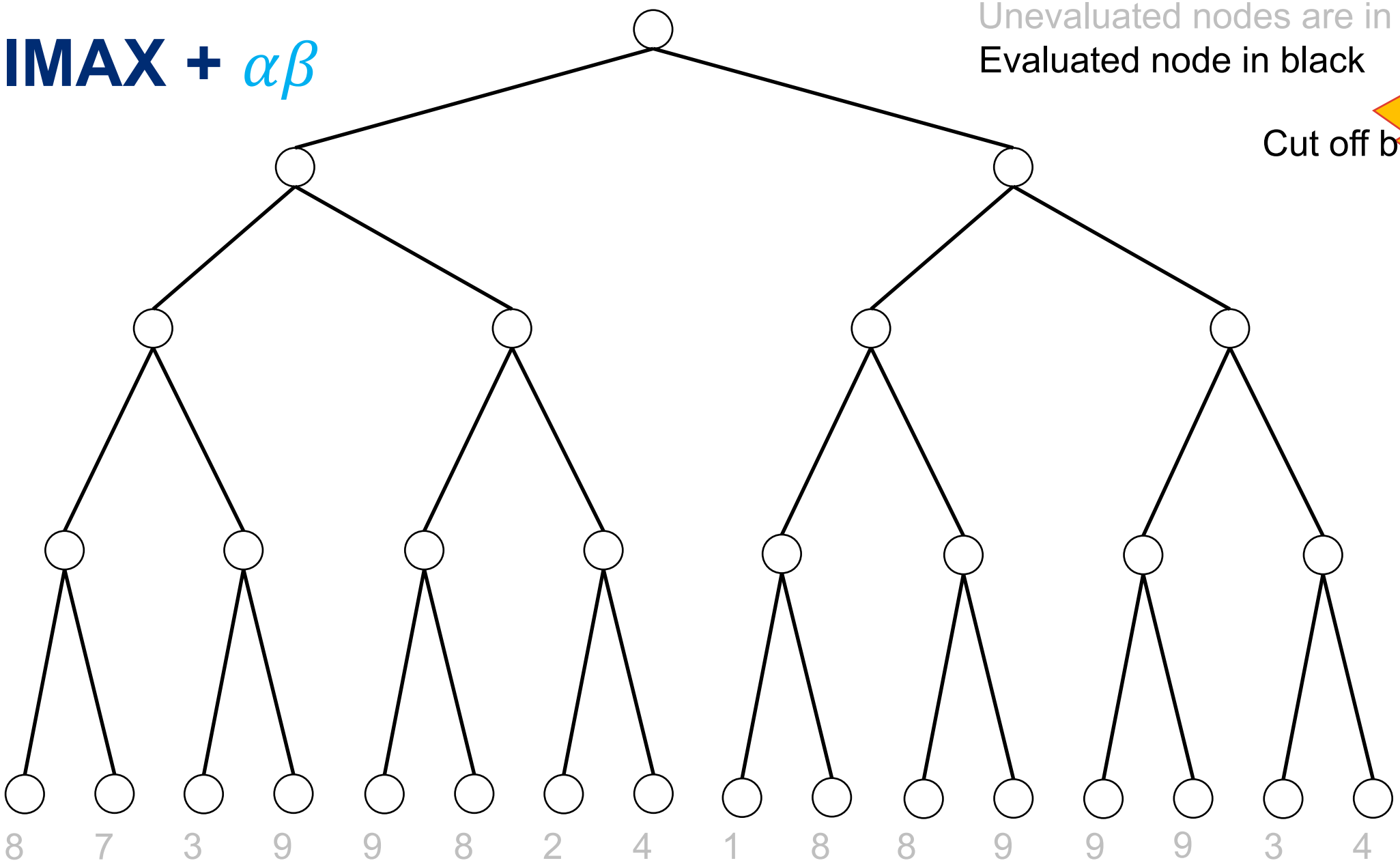
- Player 1 (MAXIMIZER) trying to maximize score
- Player 2 (MINIMIZER) trying to minimize the maximizer's score
- Game played from the perspective of the maximizer
- What should P1 do at the top node? What decision does P1 make? For instance, go Left or Right?
- Start at the left and run a static evaluation ***IF*** you need to
  - Might be able to skips certain board evaluations
  - Might be able to ignore certain branches
- Propagate the values back to the very top (the MAX node)

MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

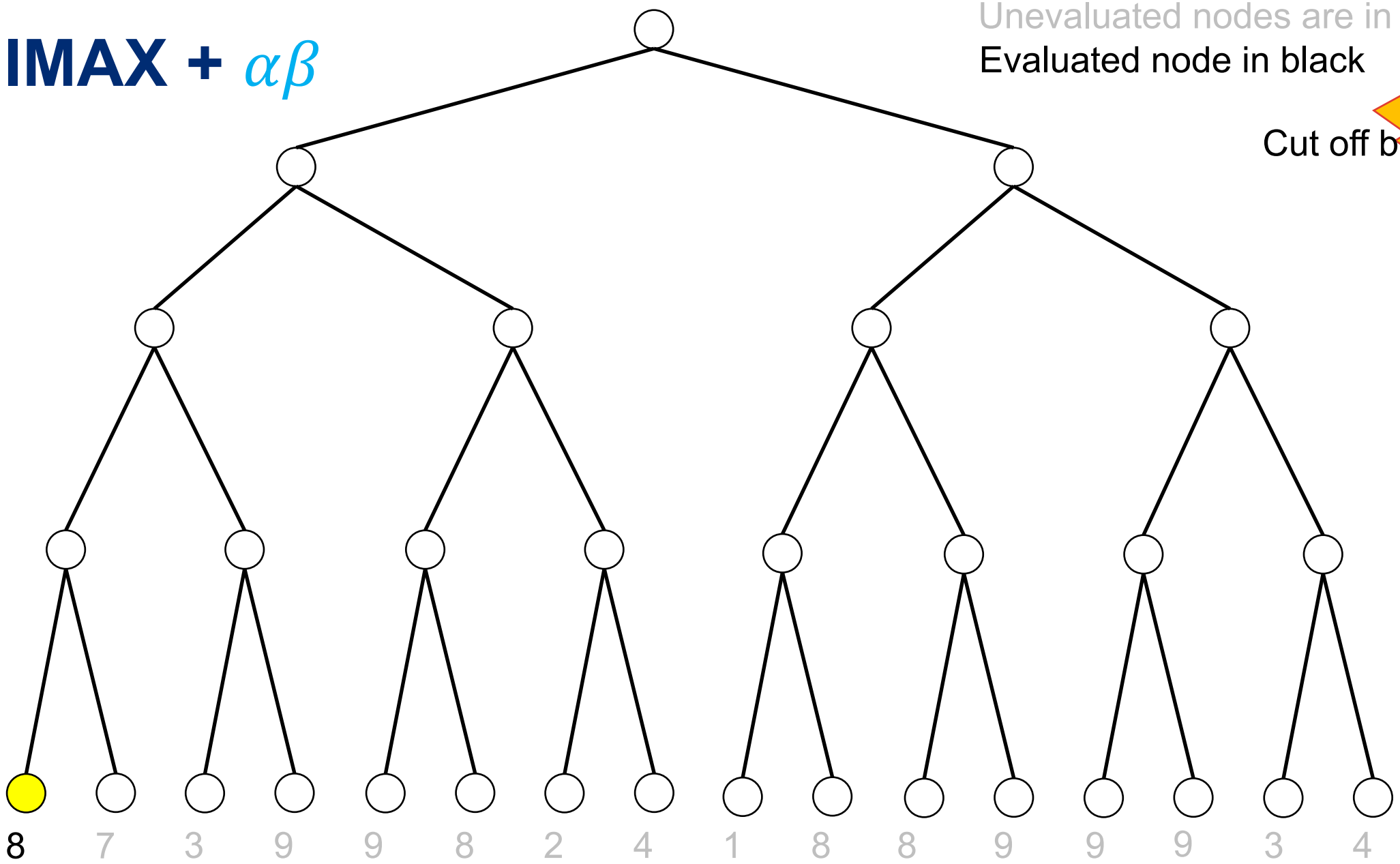


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

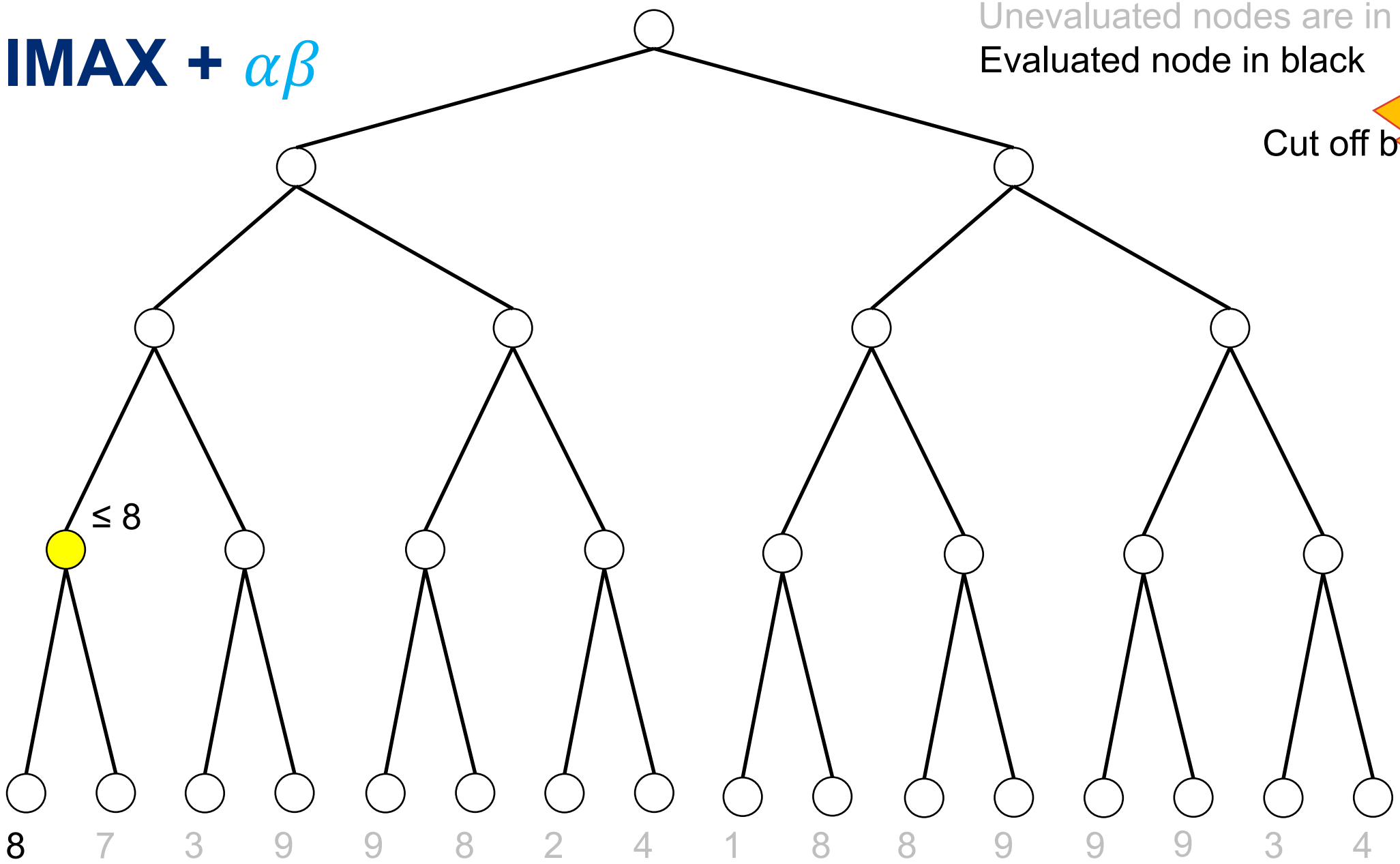


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

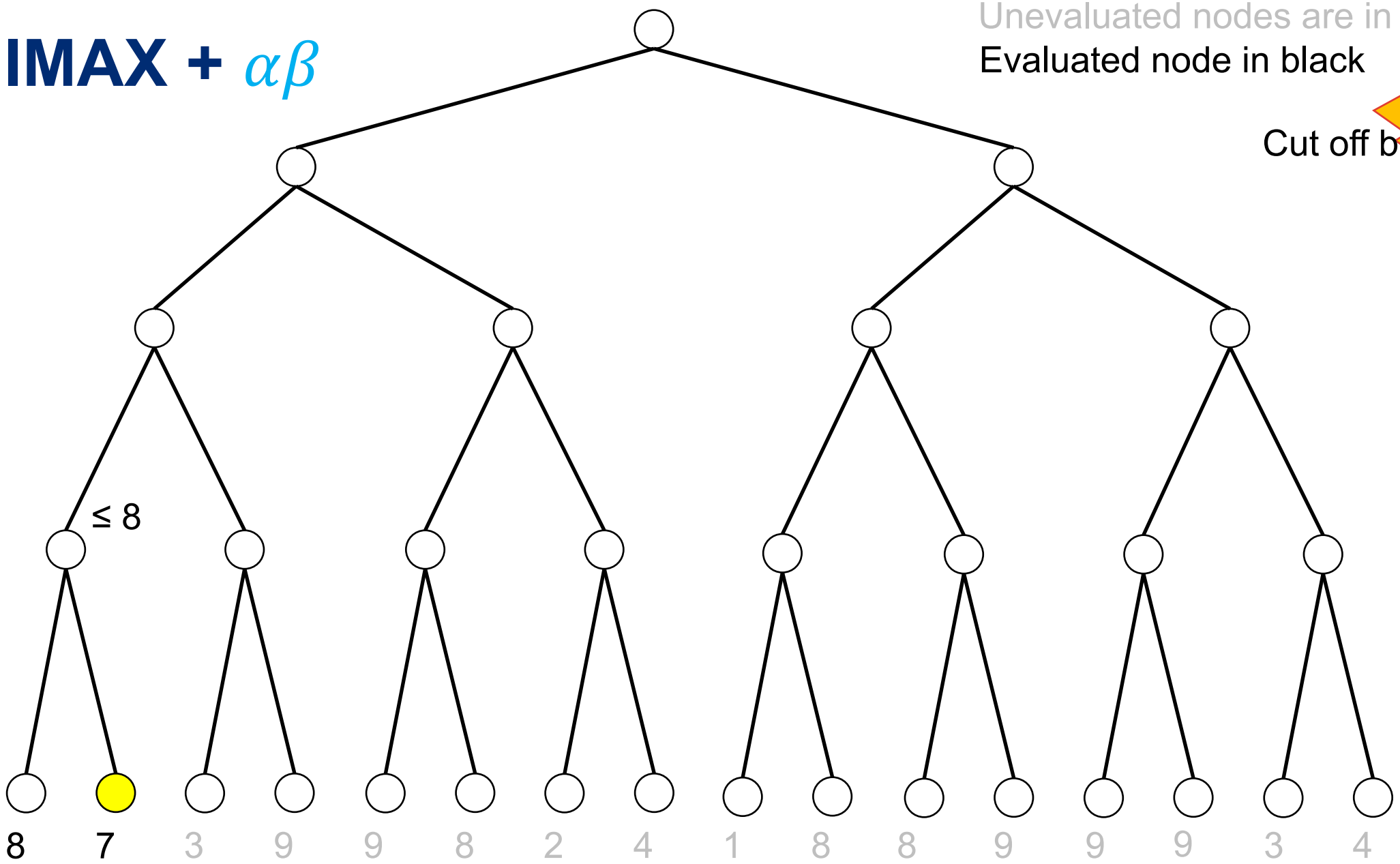


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

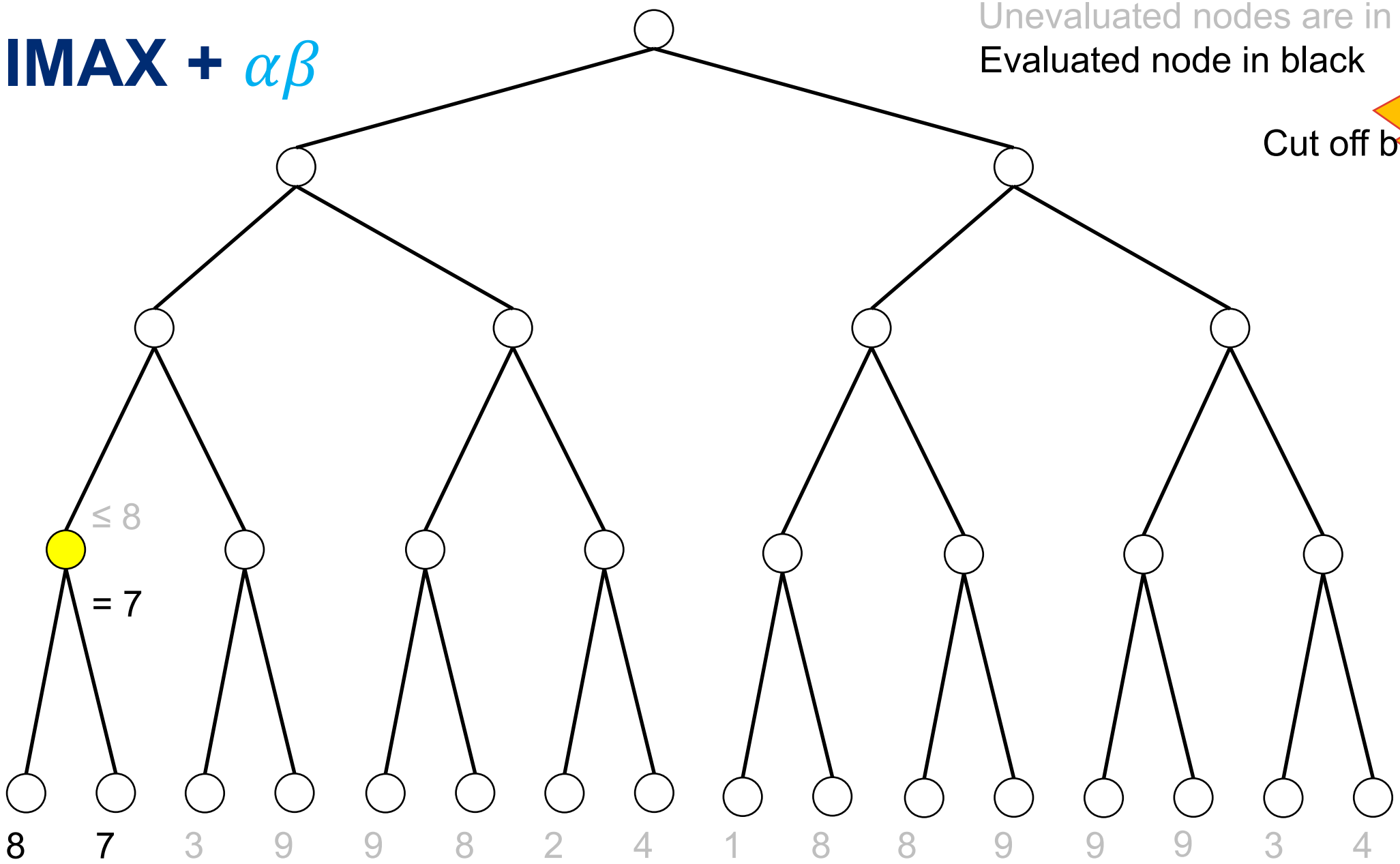


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

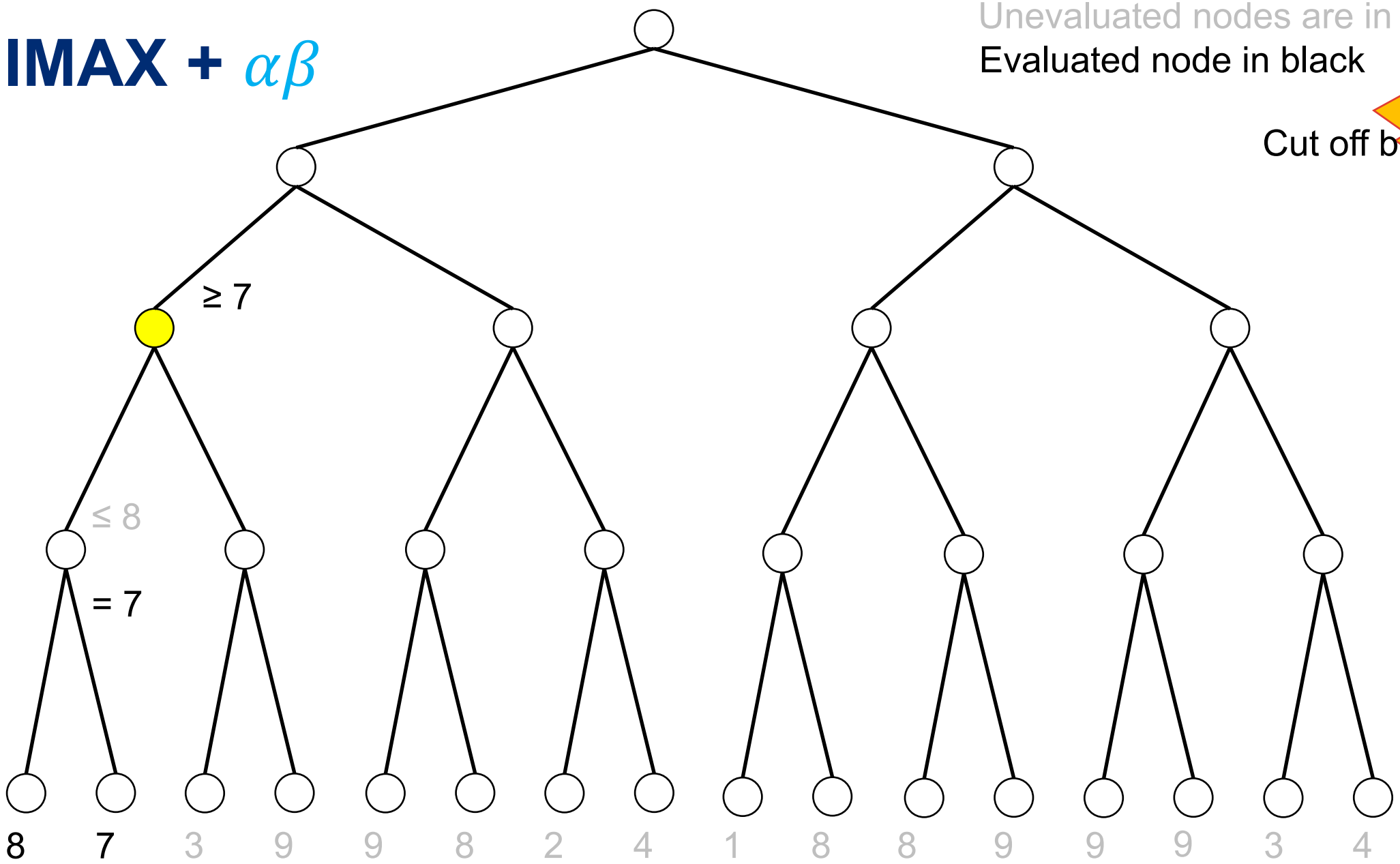


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

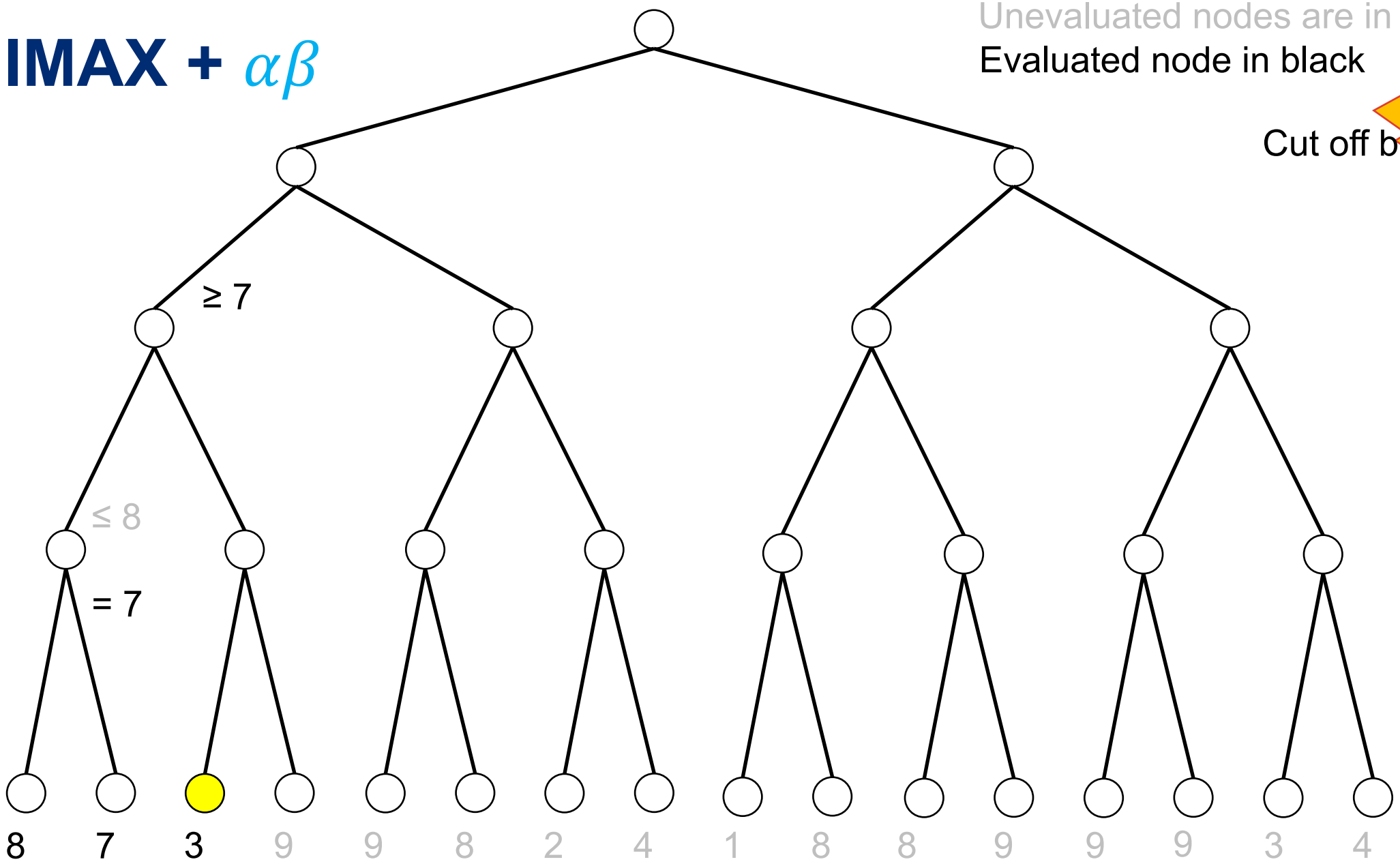


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

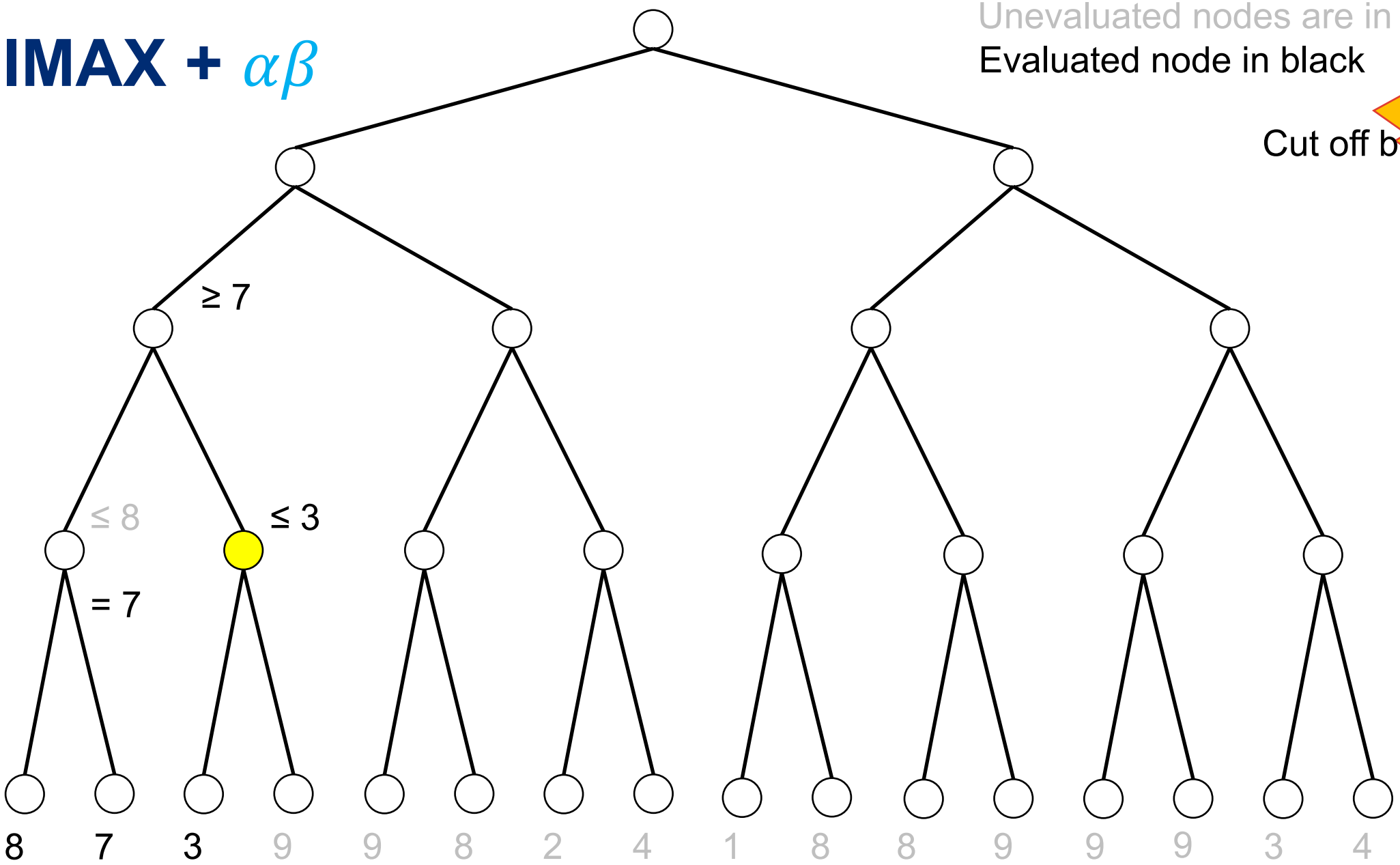


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

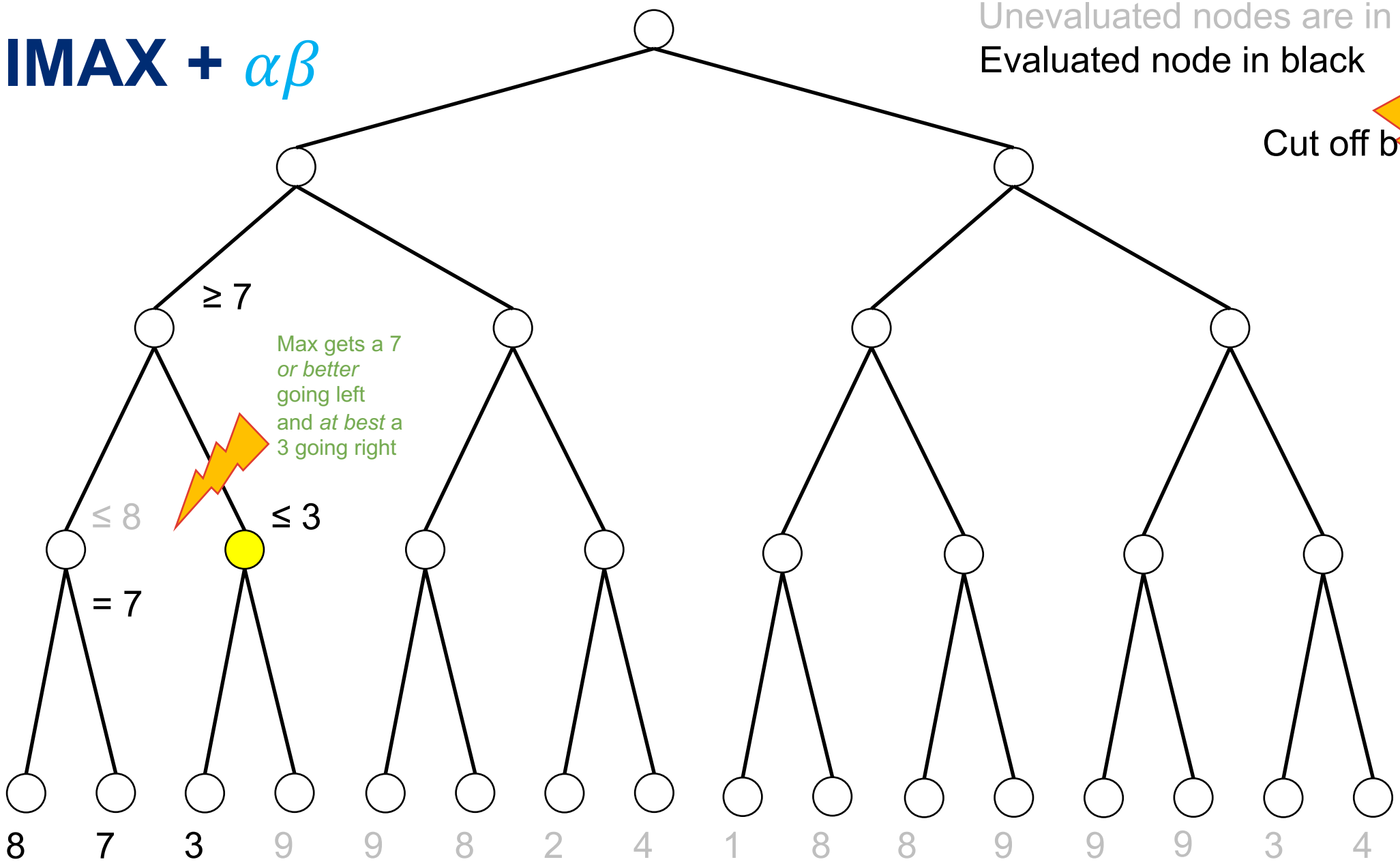


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

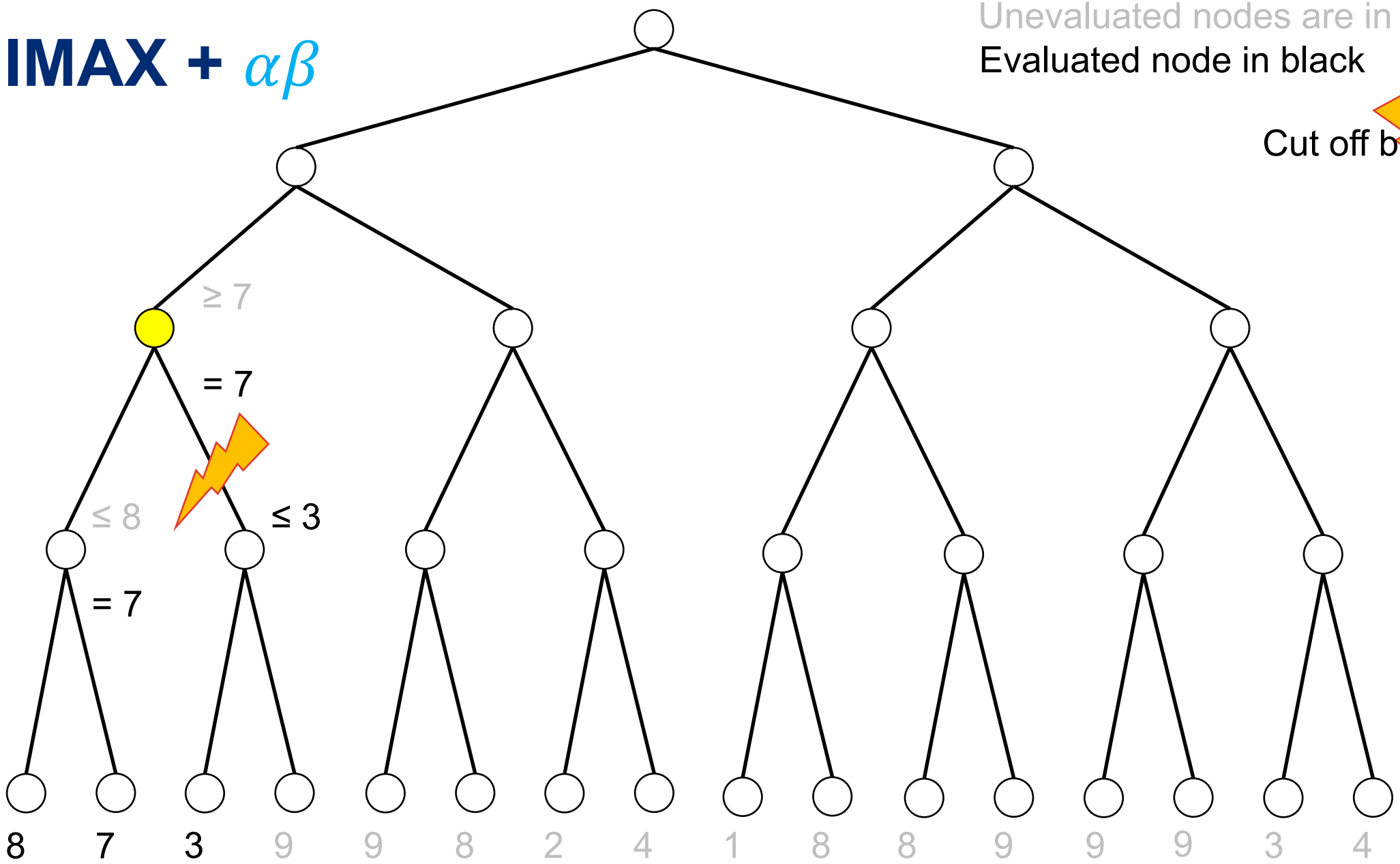


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

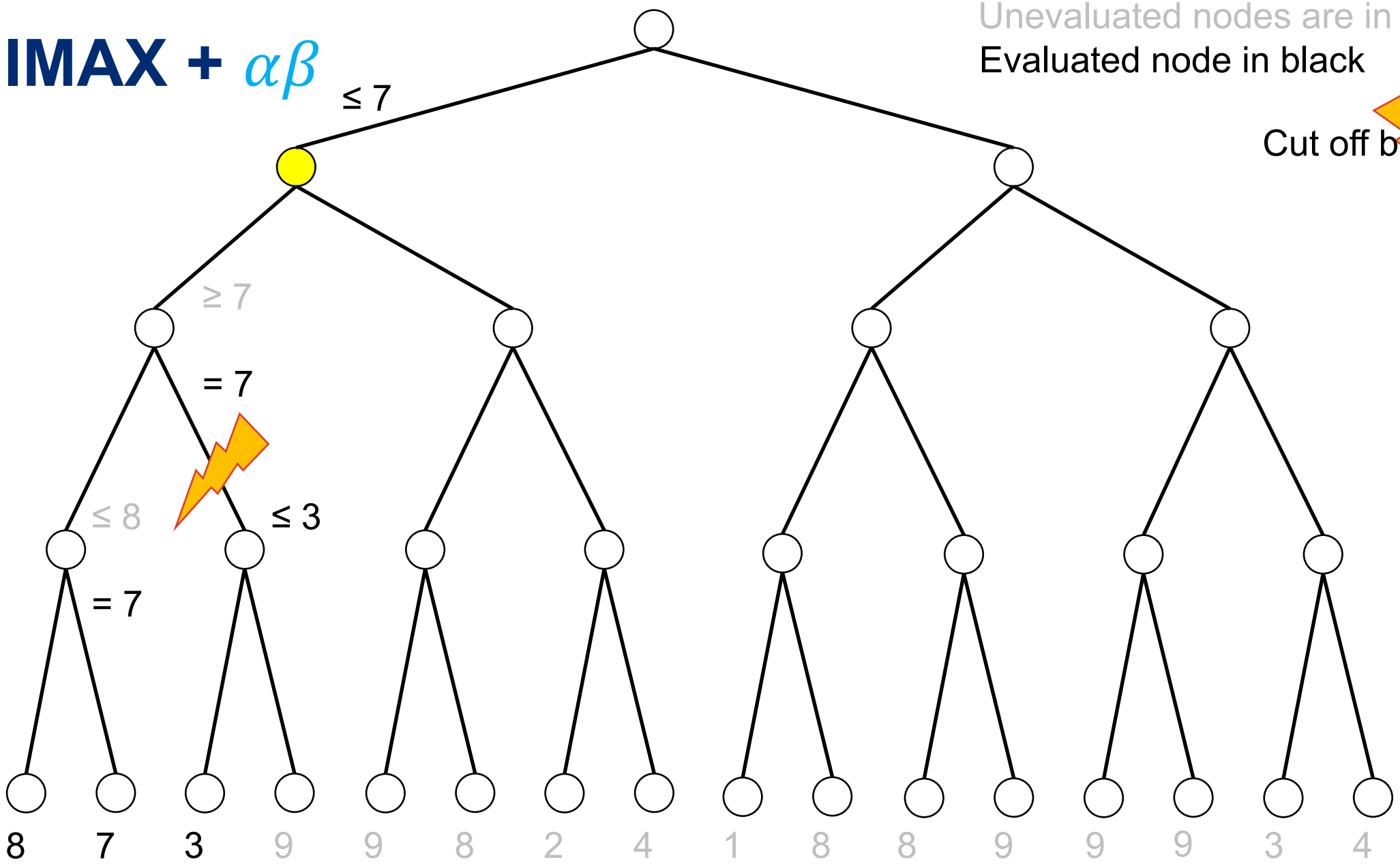


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

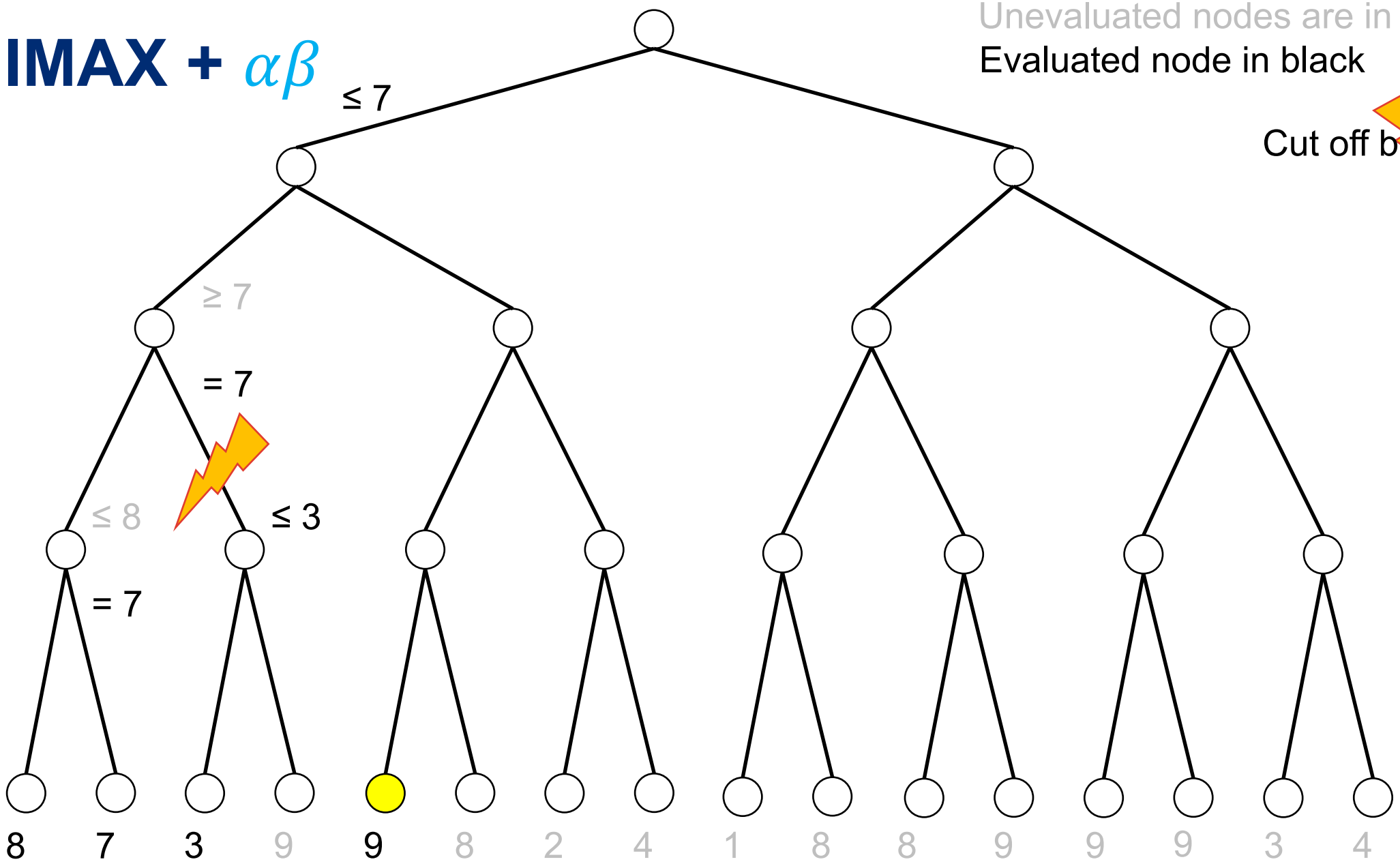


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

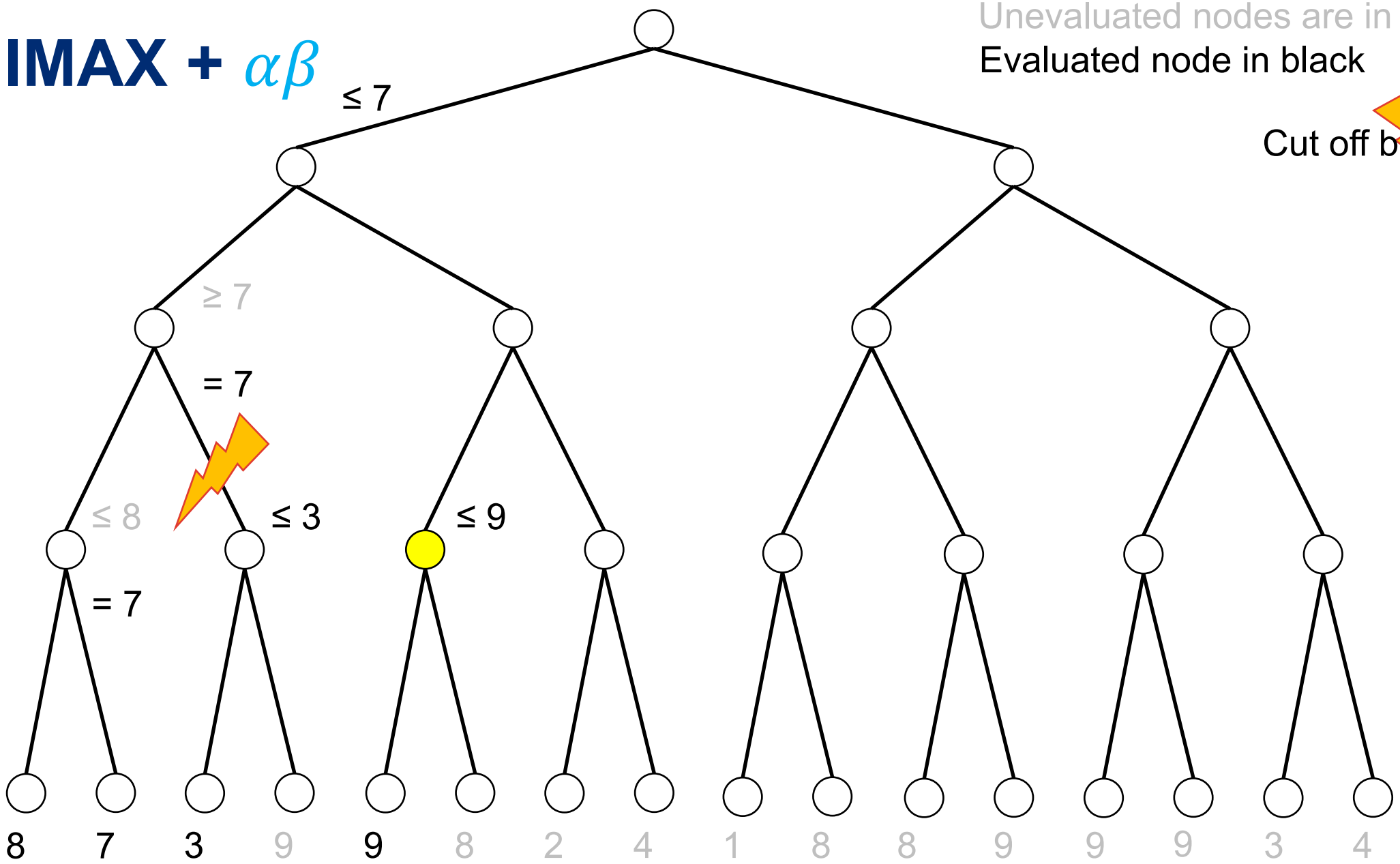


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

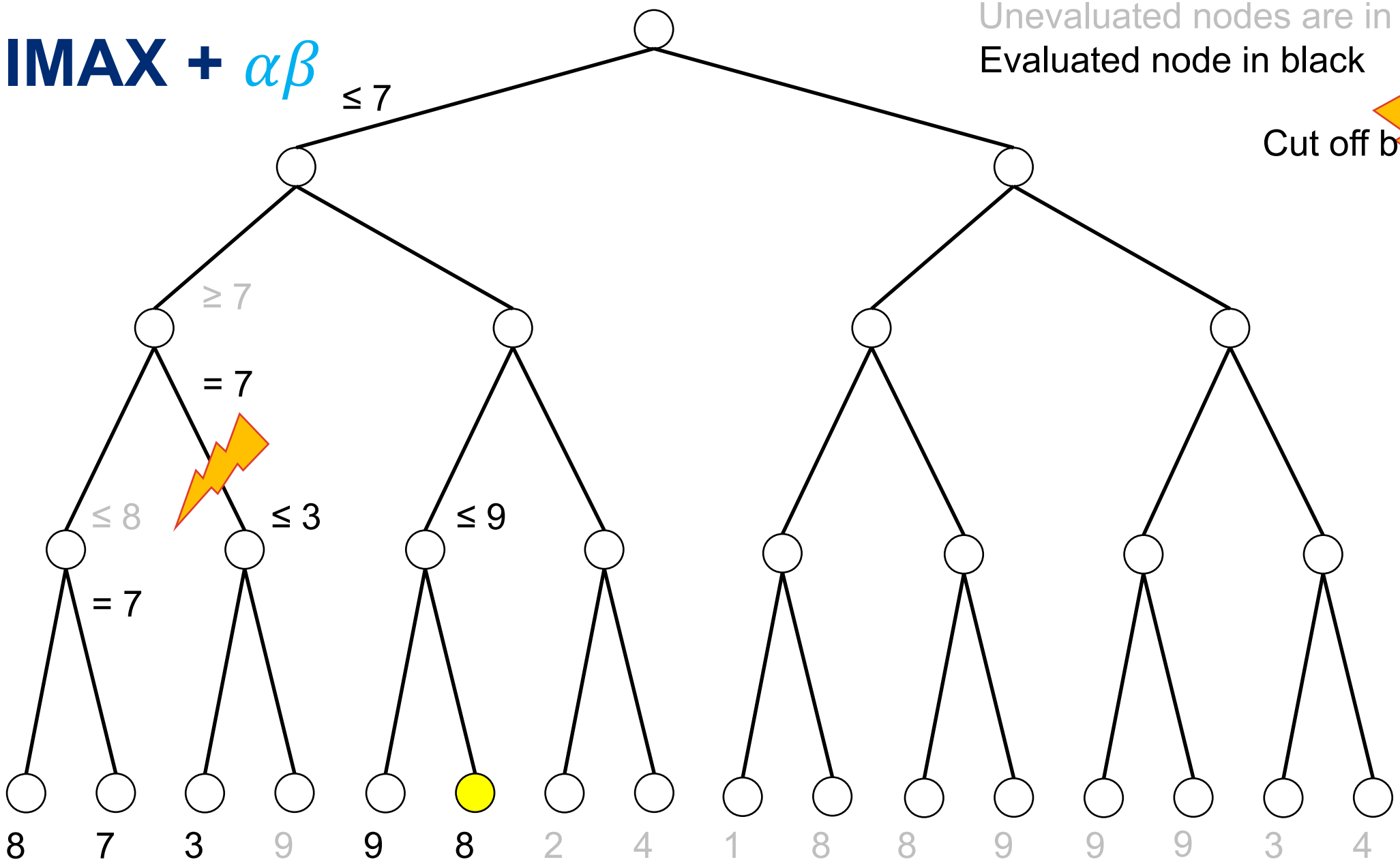


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

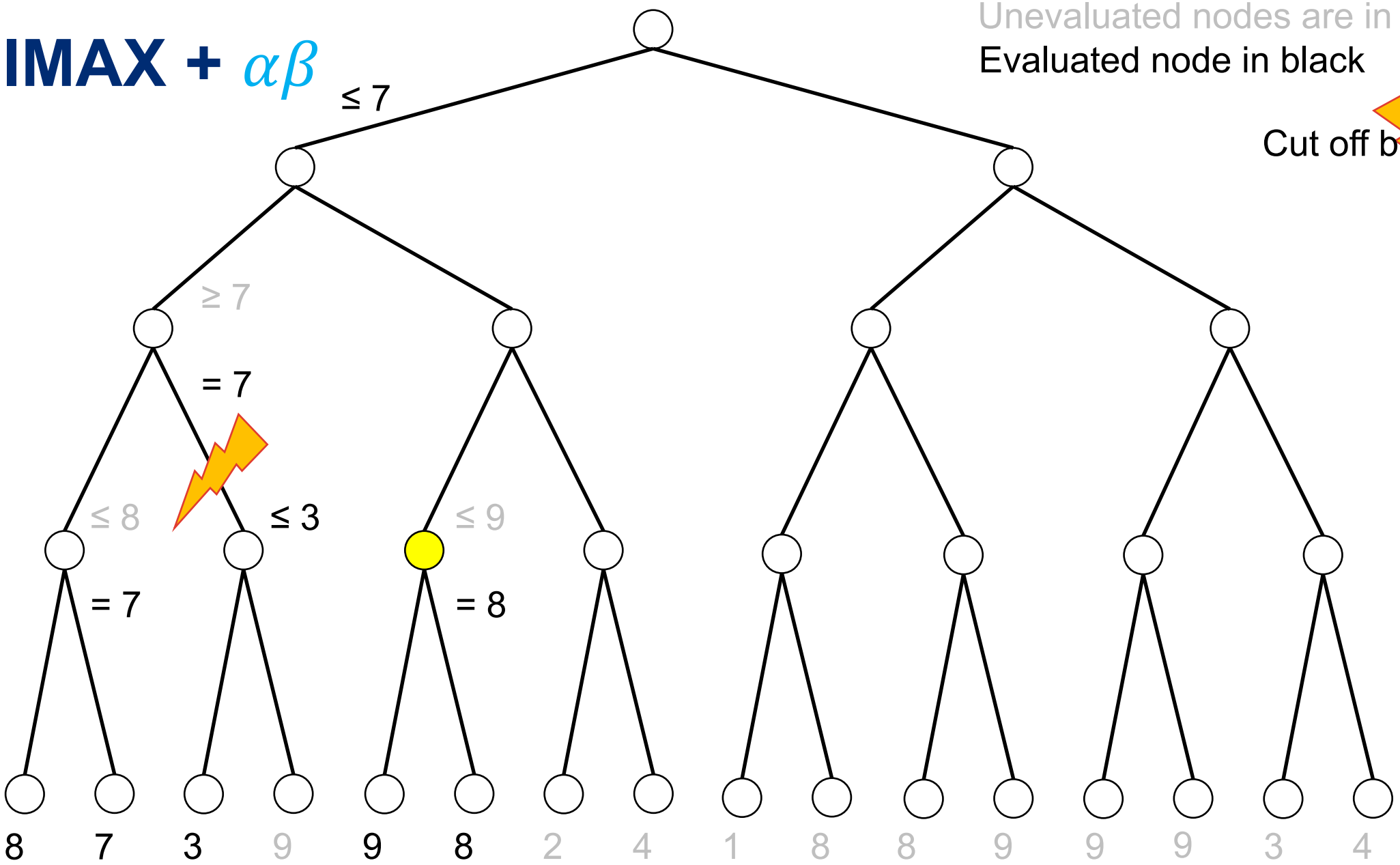


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

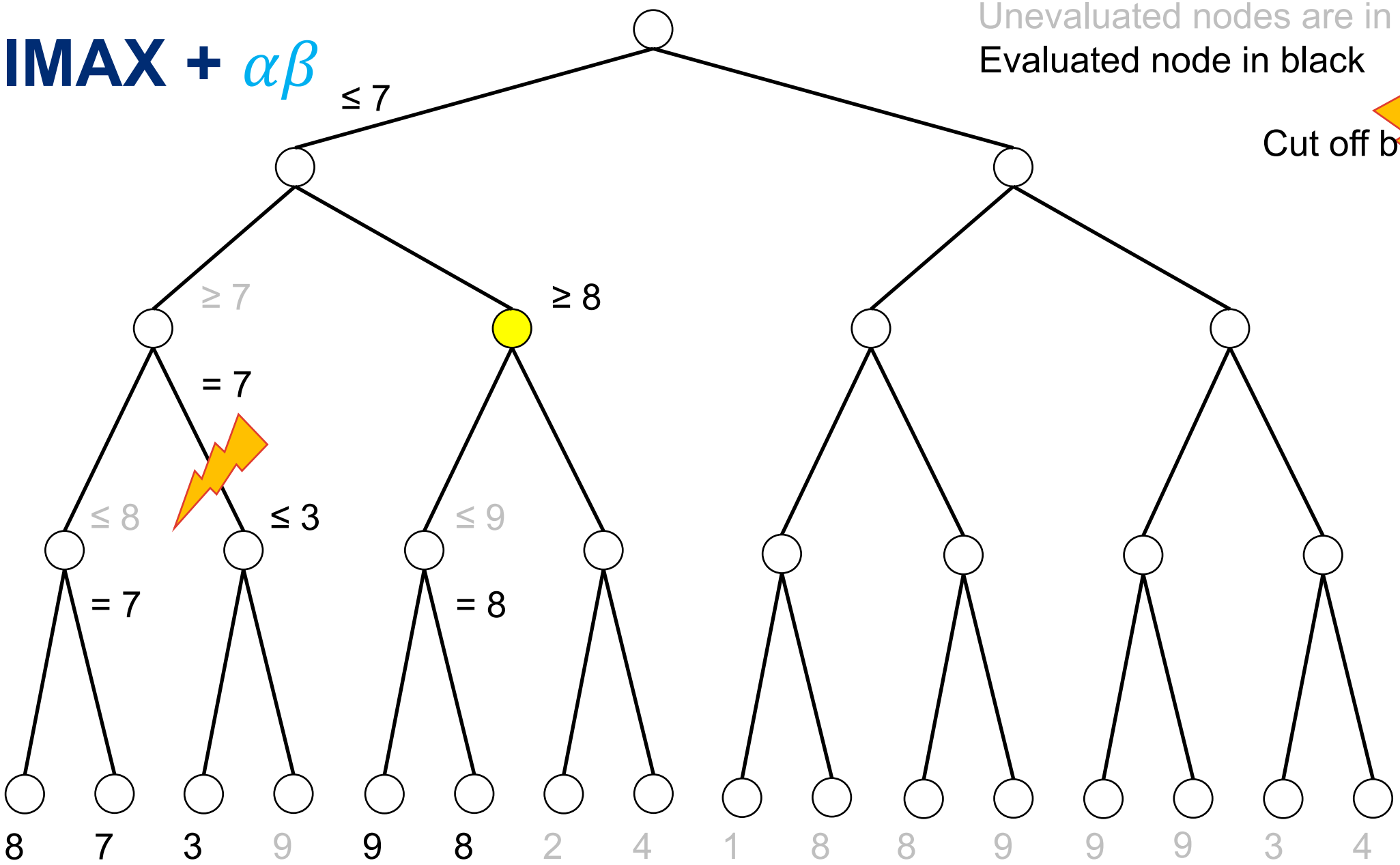


MINIMAX + $\alpha\beta$

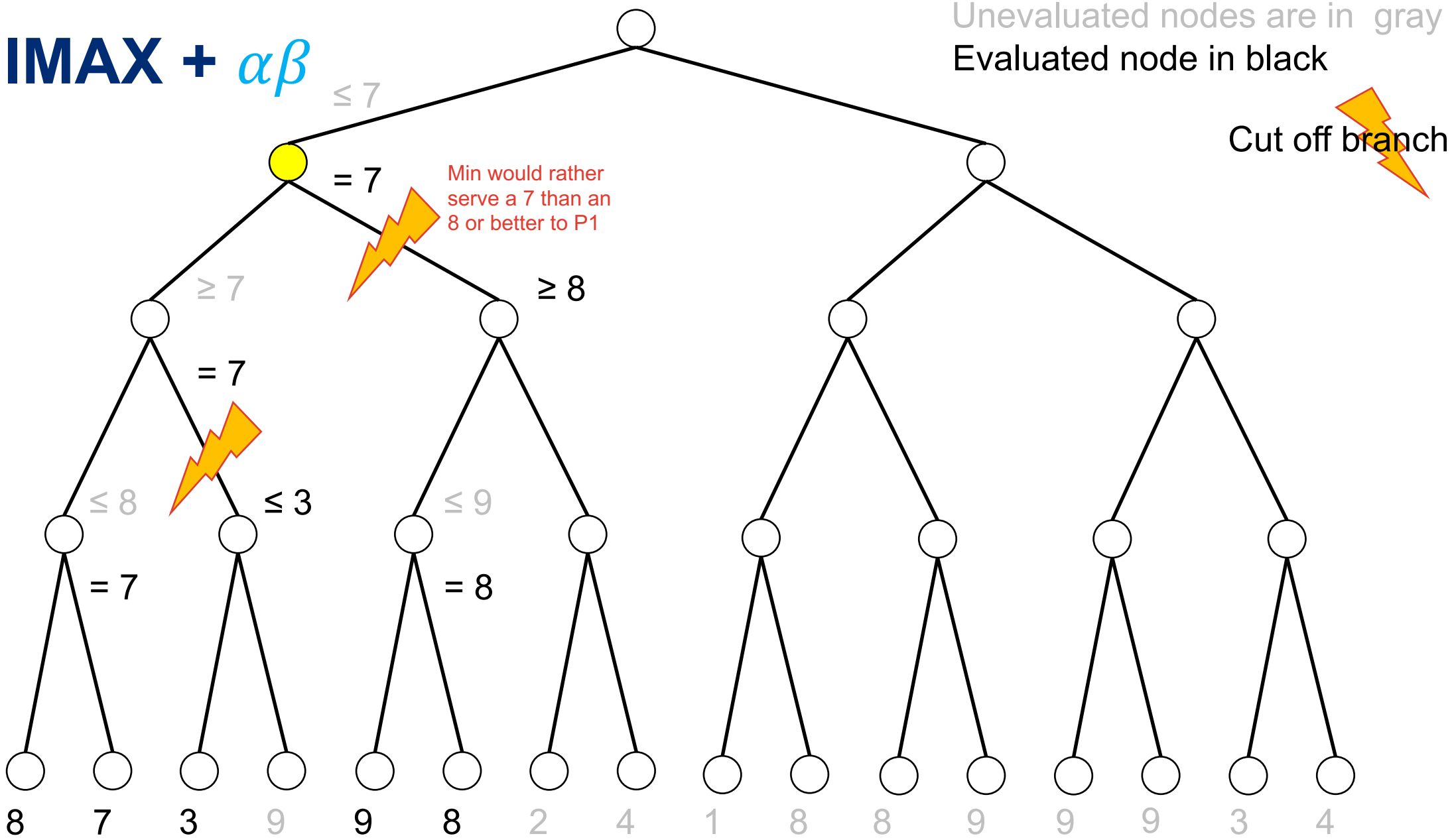
Unevaluated nodes are in gray

Evaluated node in black

Cut off branch



MINIMAX + $\alpha\beta$

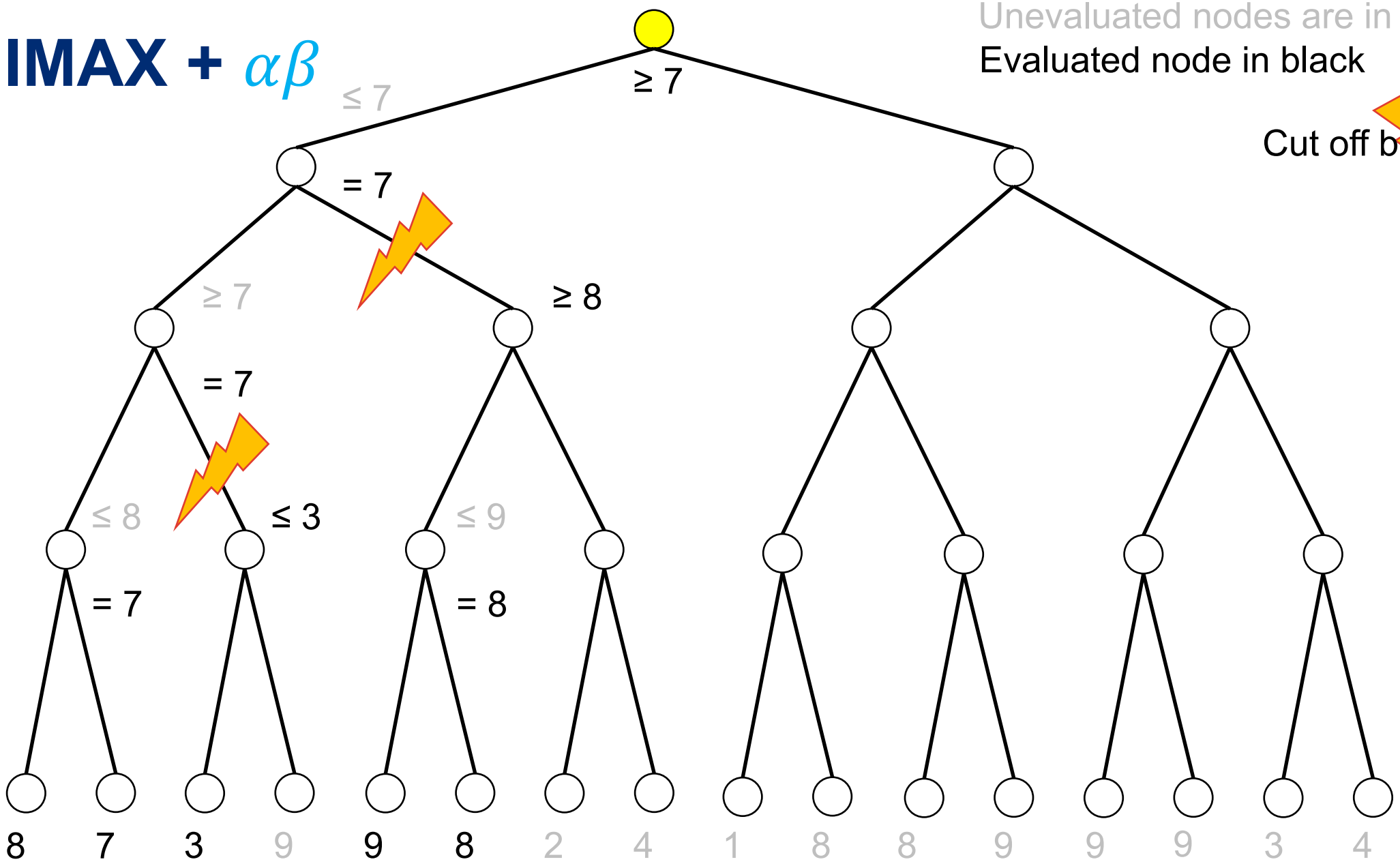


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

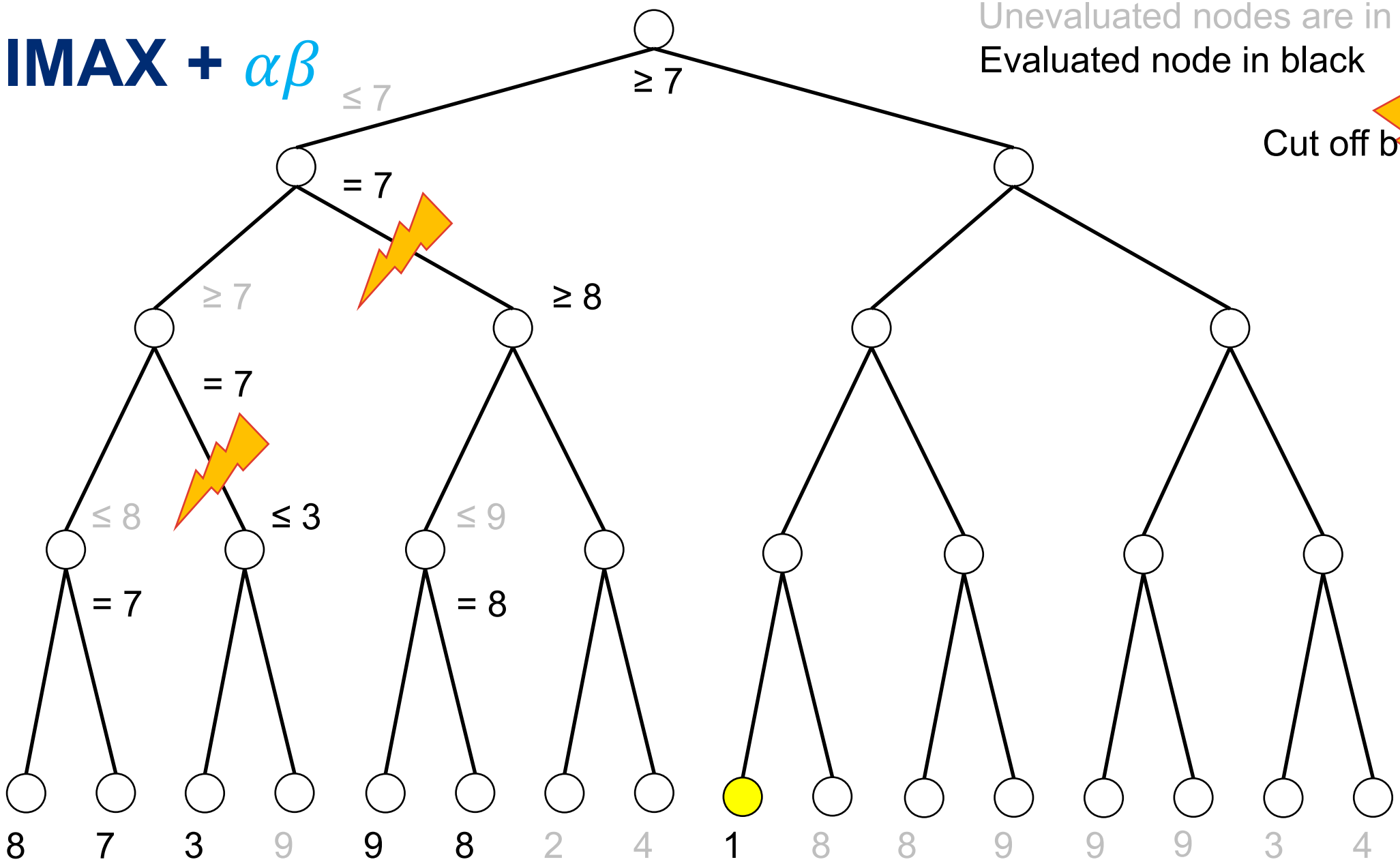


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

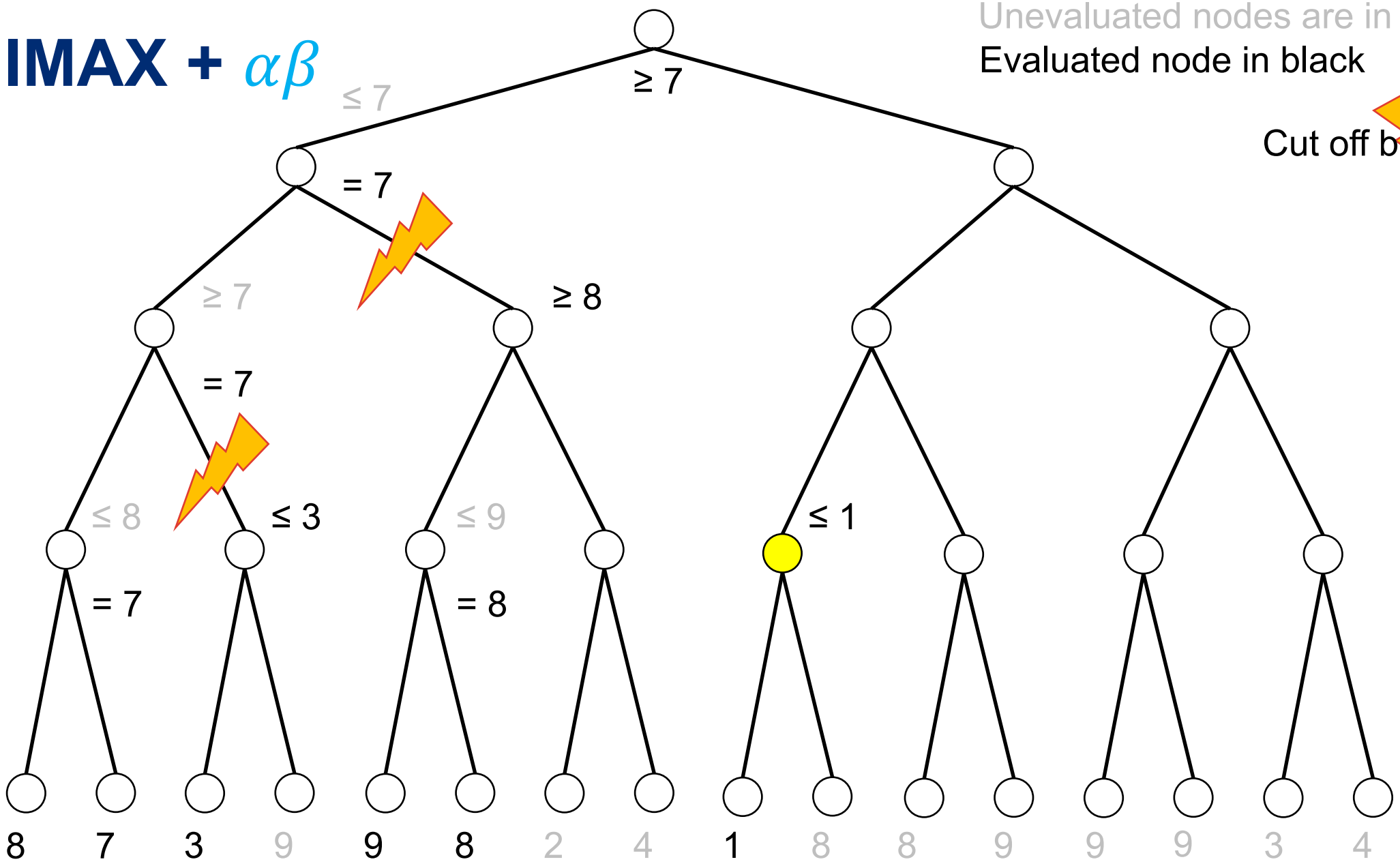


MINIMAX + $\alpha\beta$

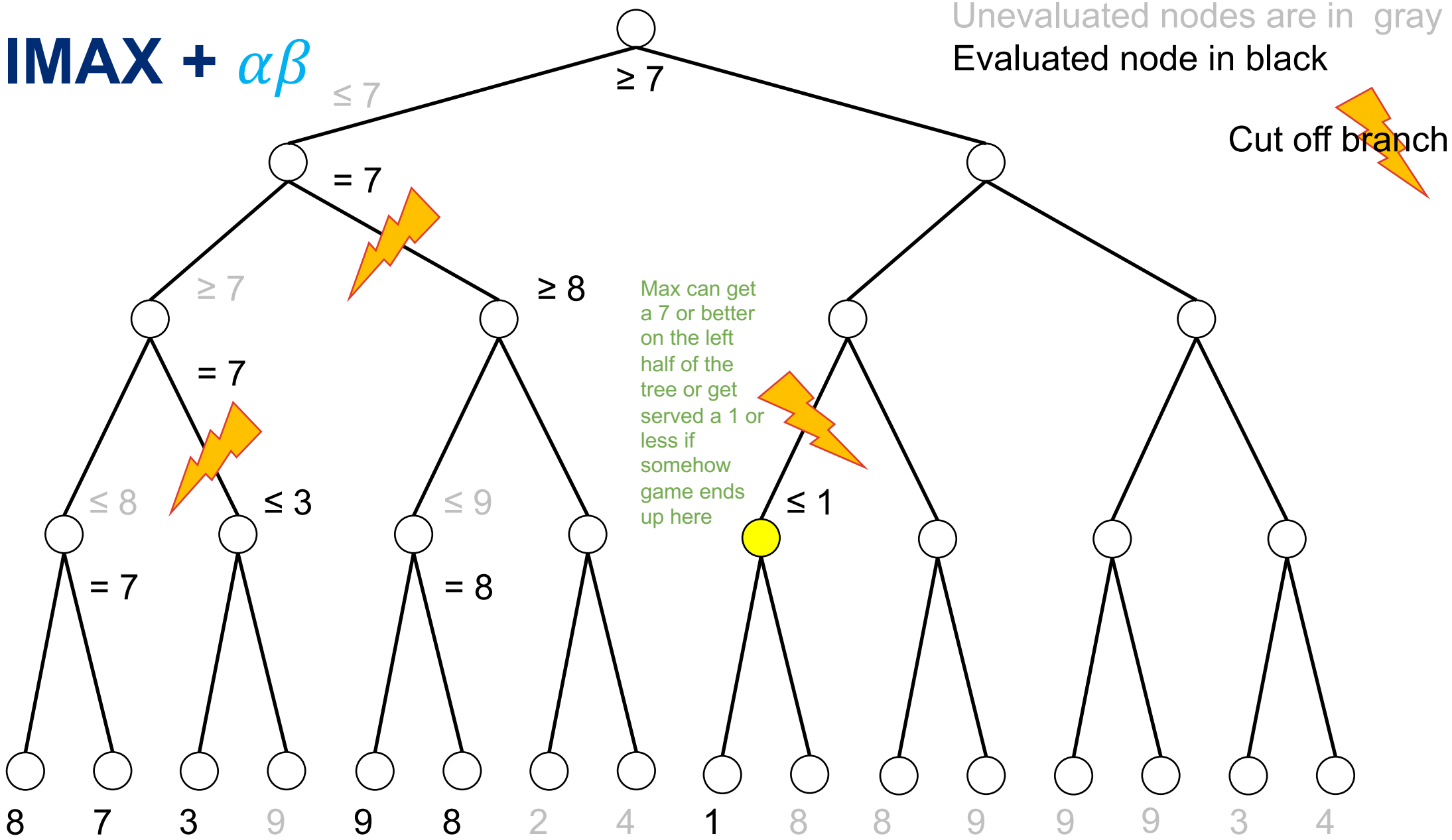
Unevaluated nodes are in gray

Evaluated node in black

Cut off branch



MINIMAX + $\alpha\beta$

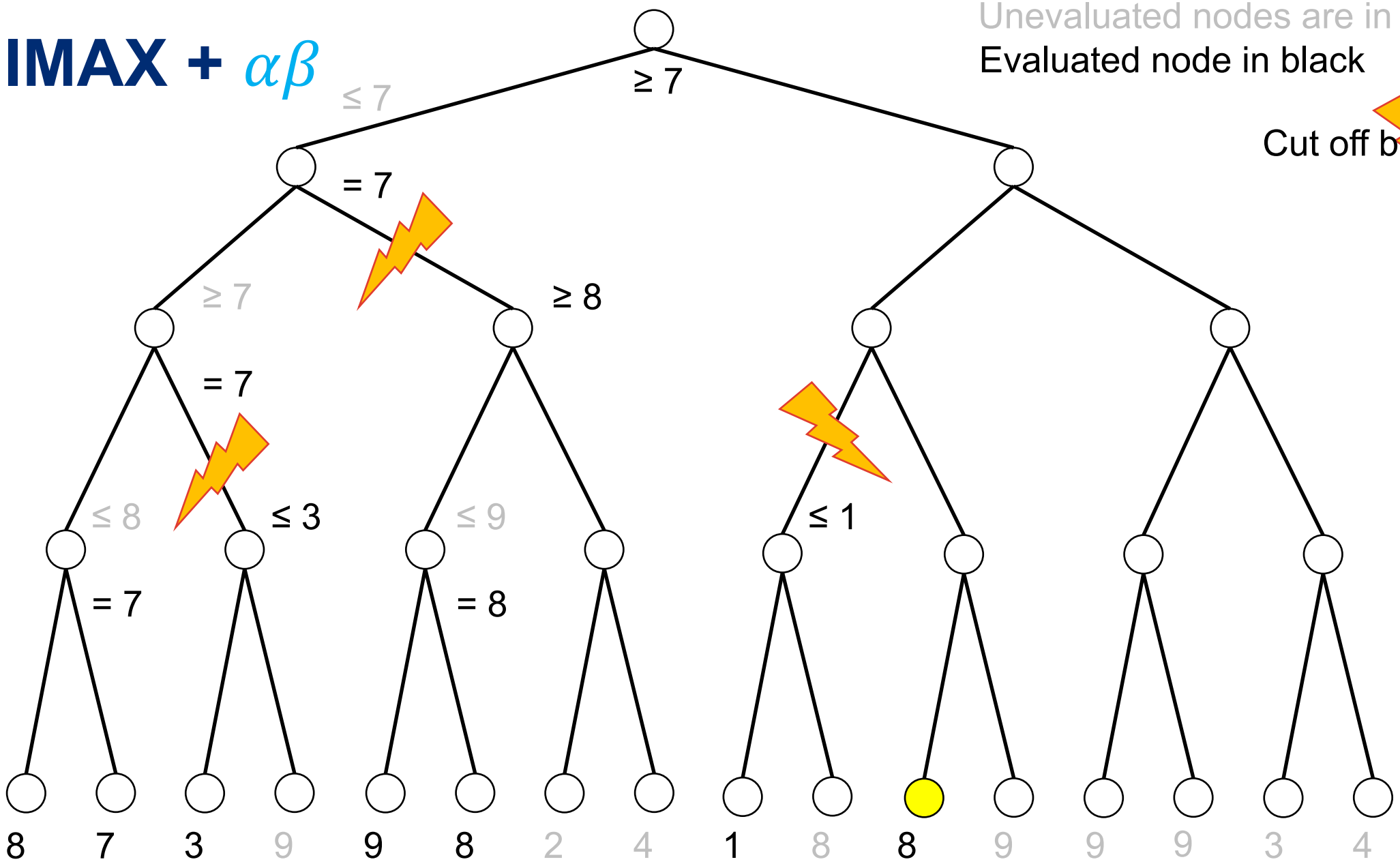


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

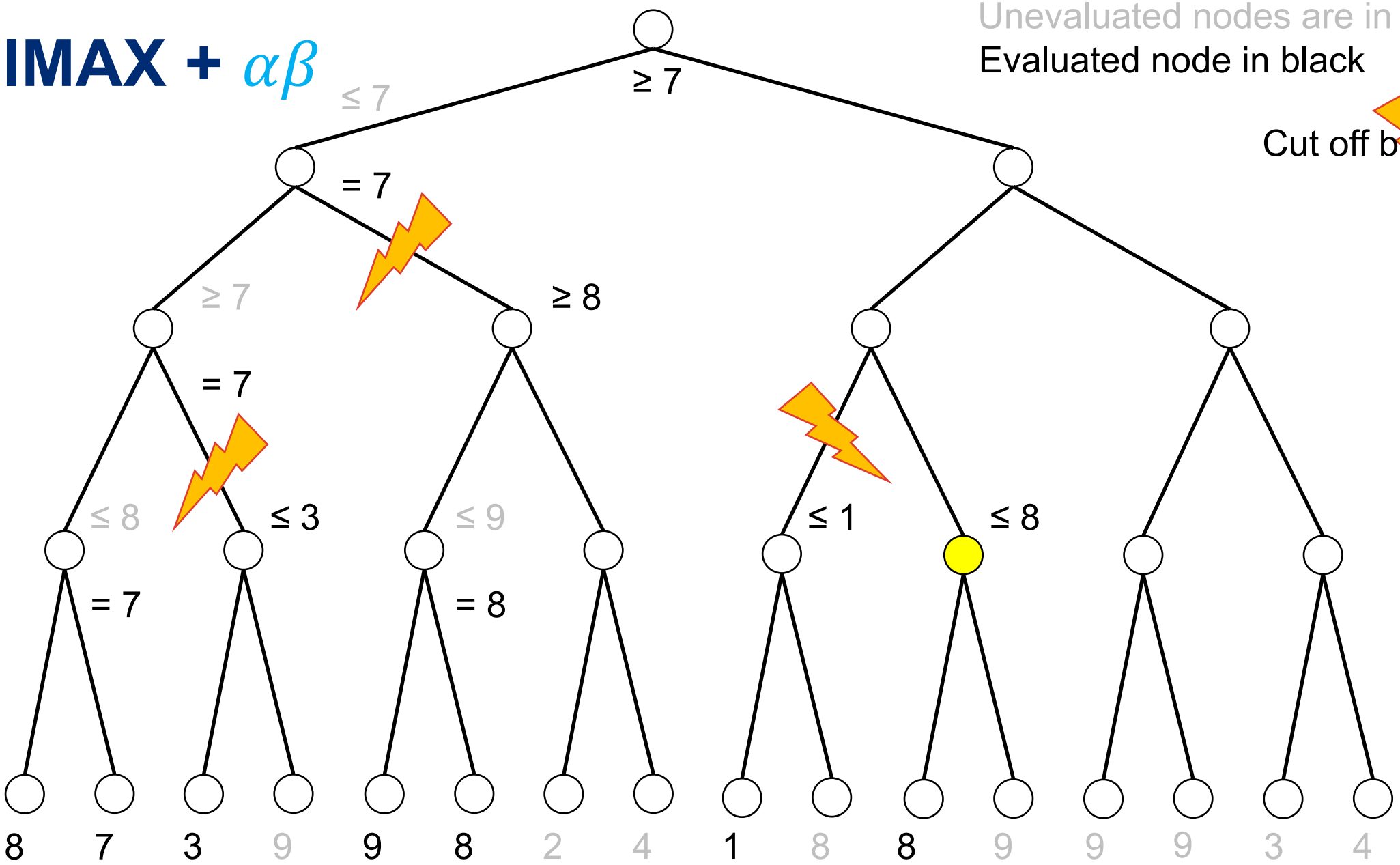


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

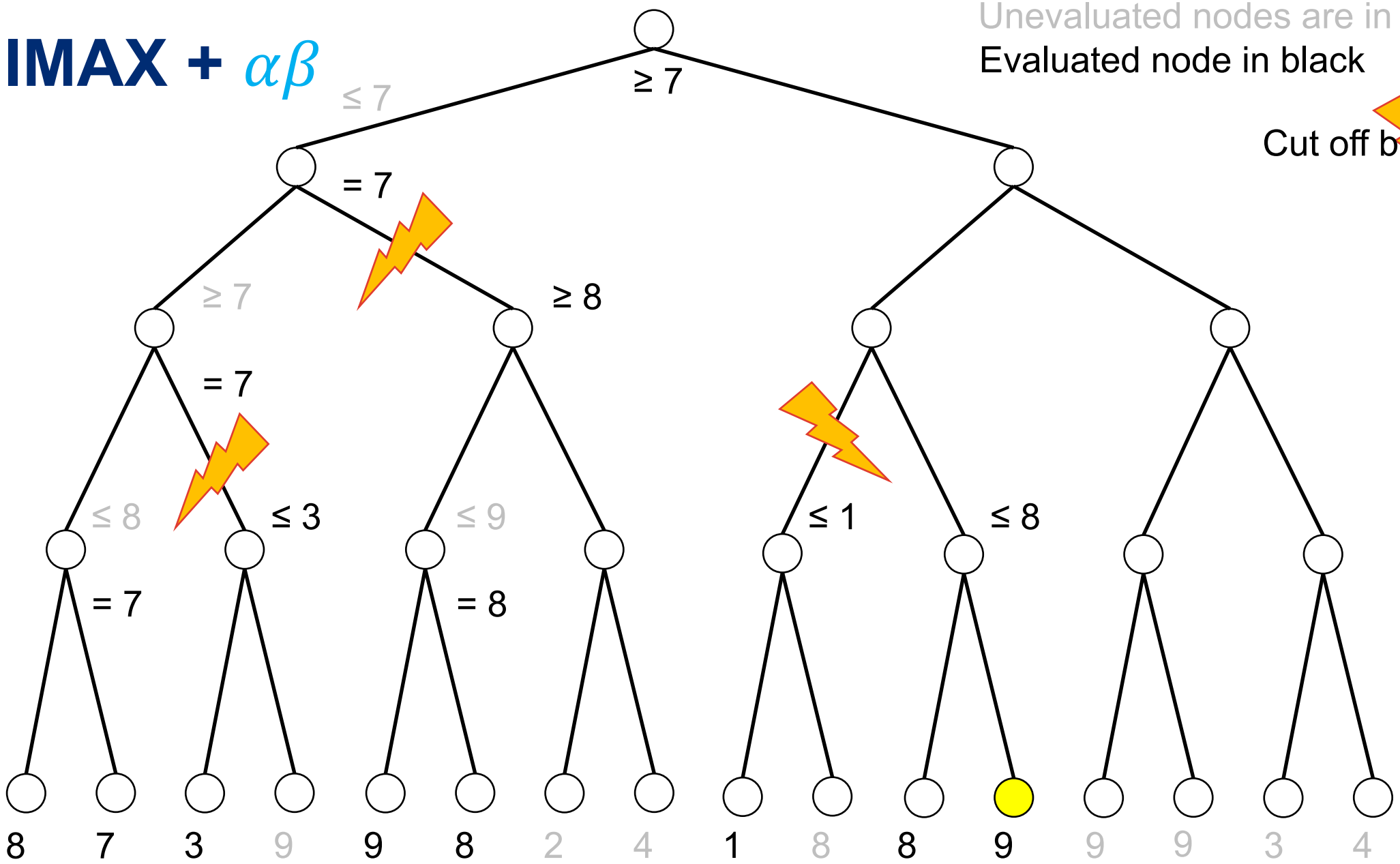


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

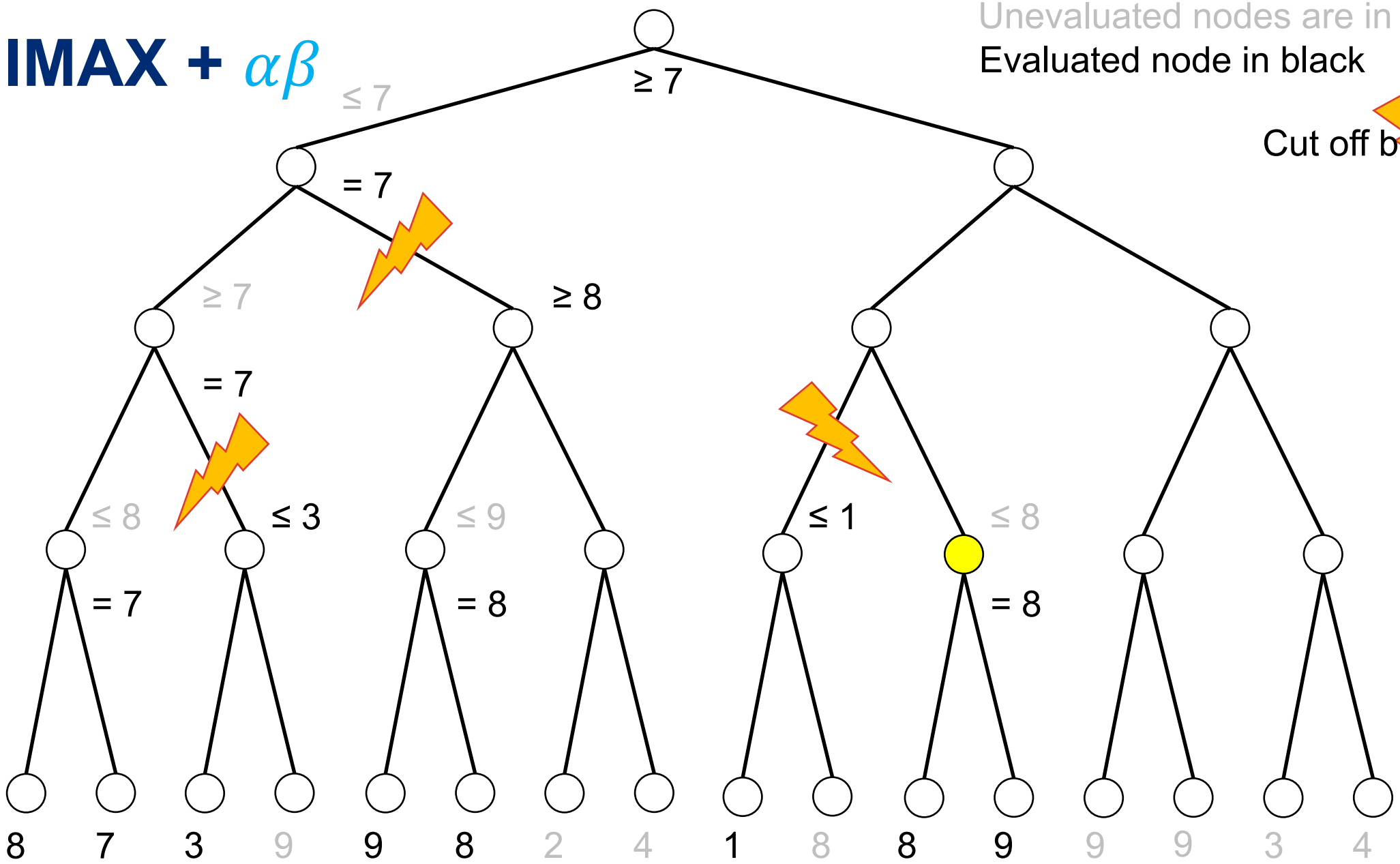


MINIMAX + $\alpha\beta$

Unevaluated nodes are in gray

Evaluated node in black

Cut off branch

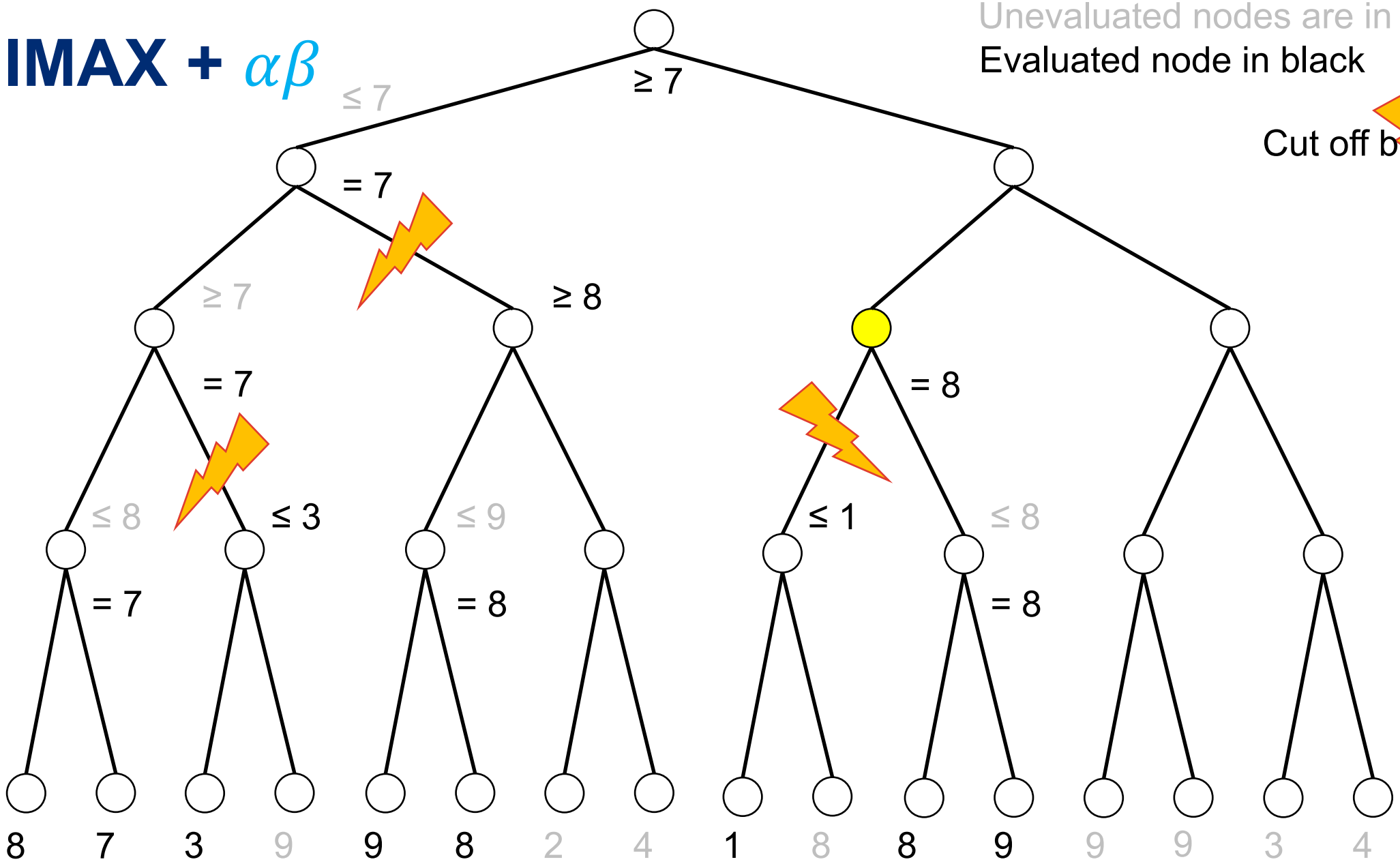


MINIMAX + $\alpha\beta$

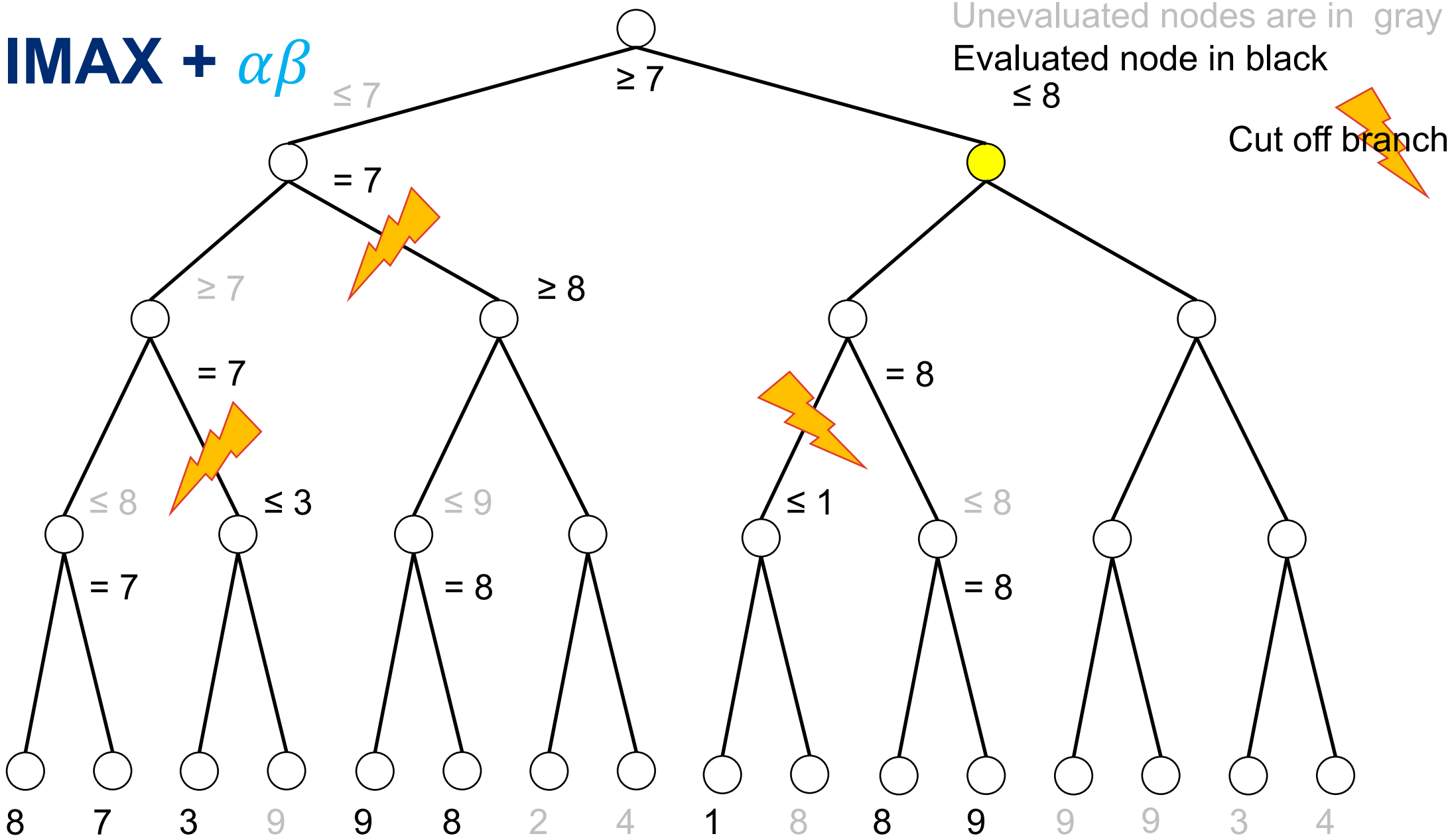
Unevaluated nodes are in gray

Evaluated node in black

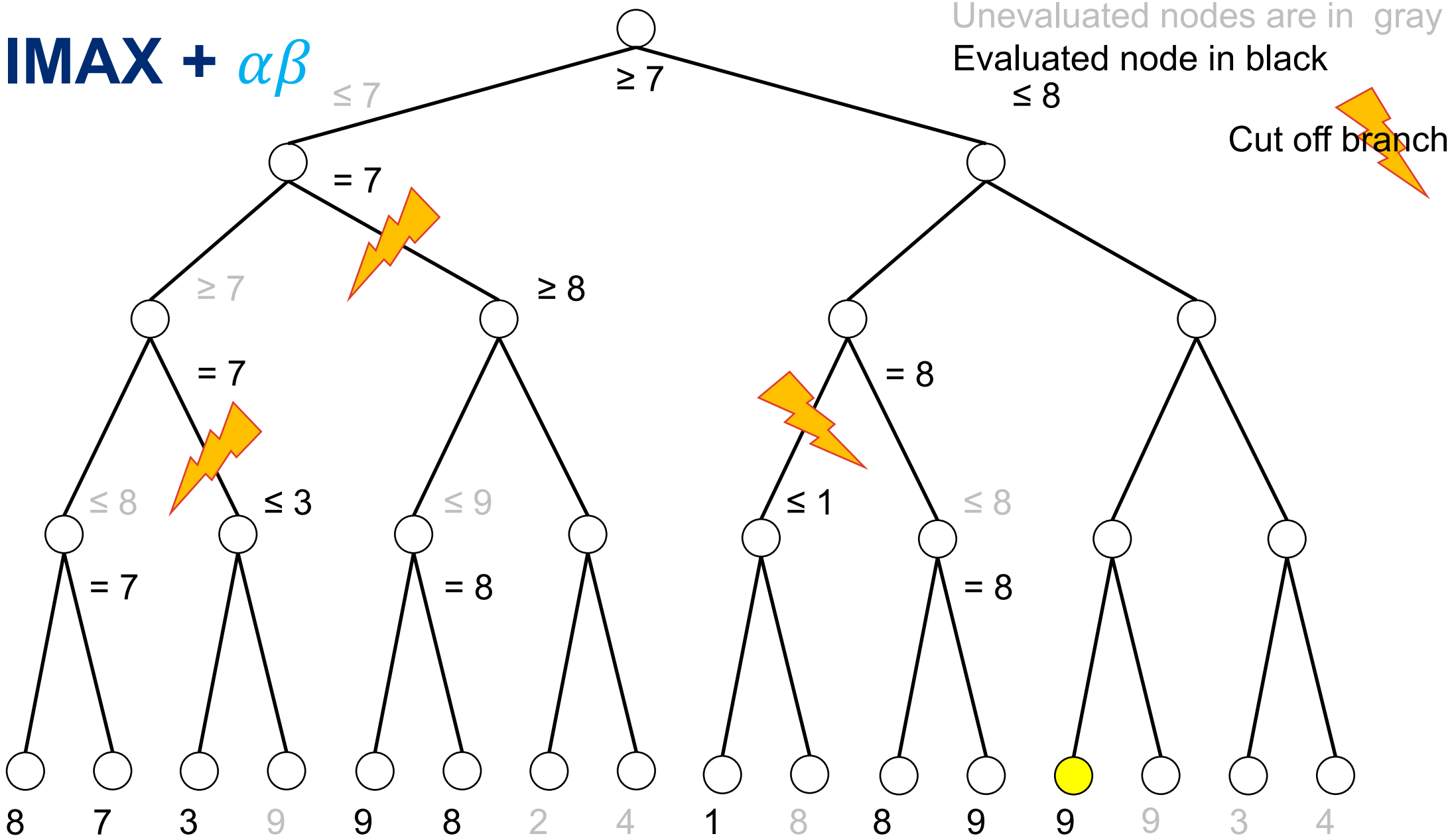
Cut off branch



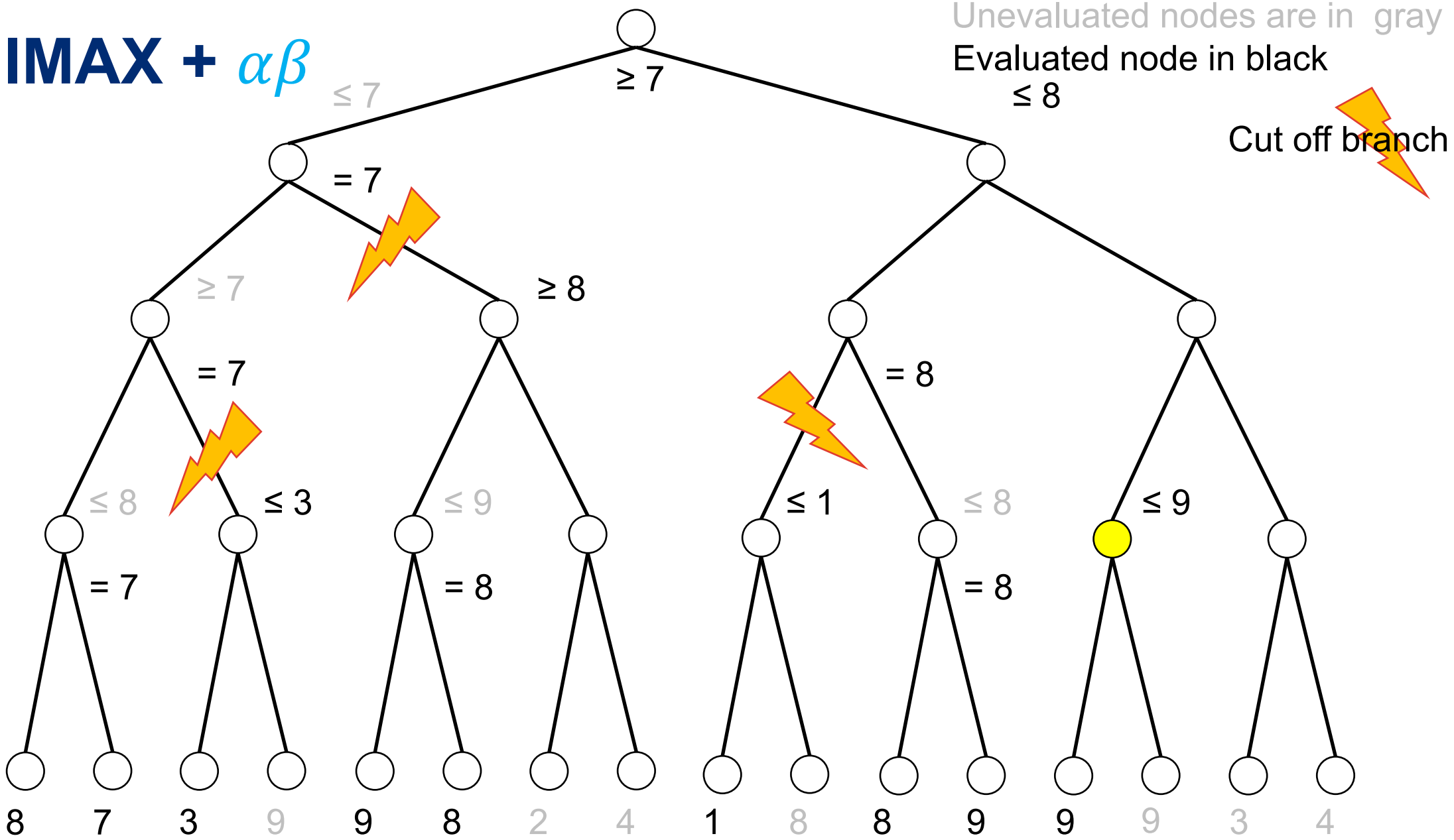
MINIMAX + $\alpha\beta$



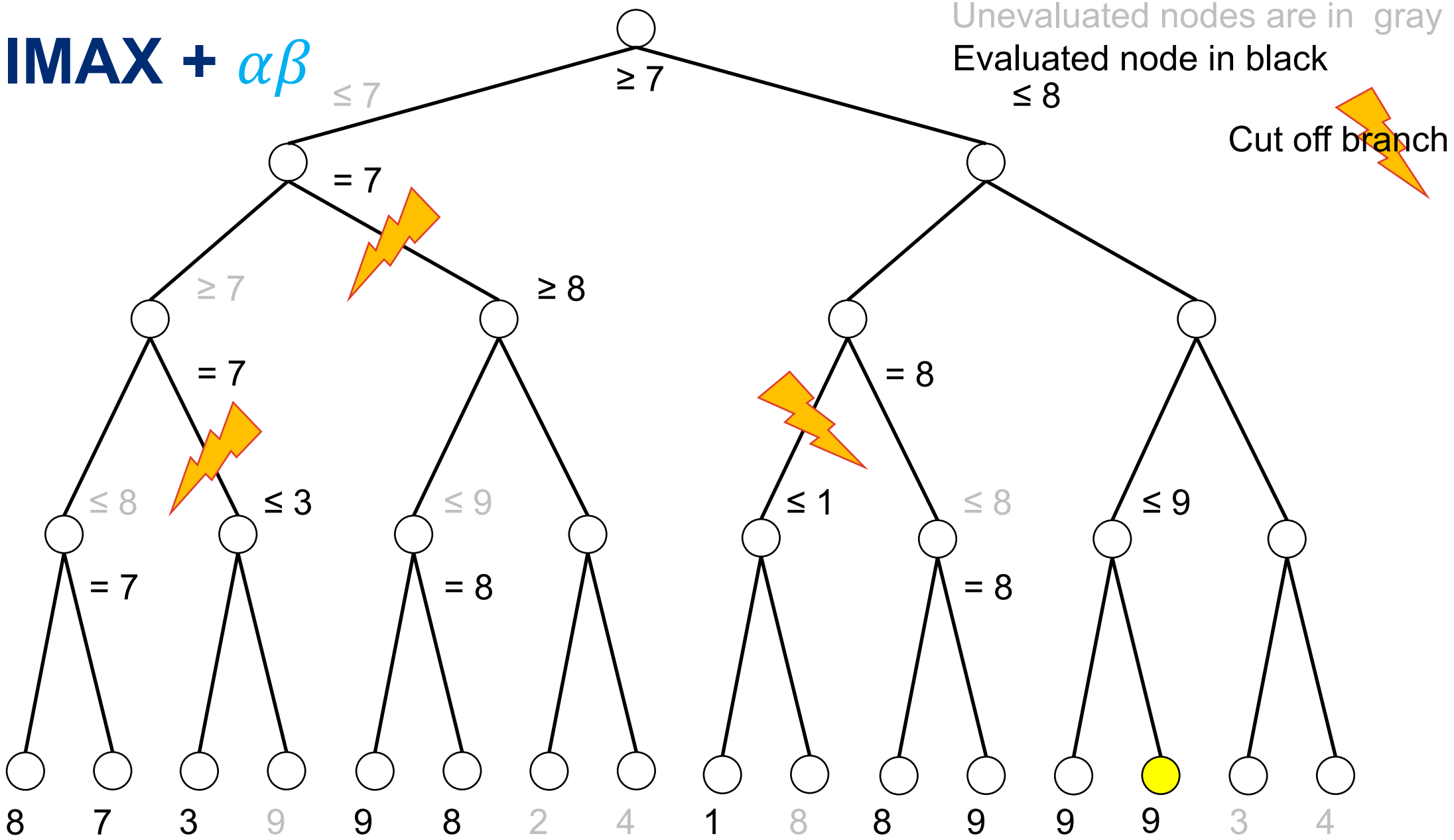
MINIMAX + $\alpha\beta$



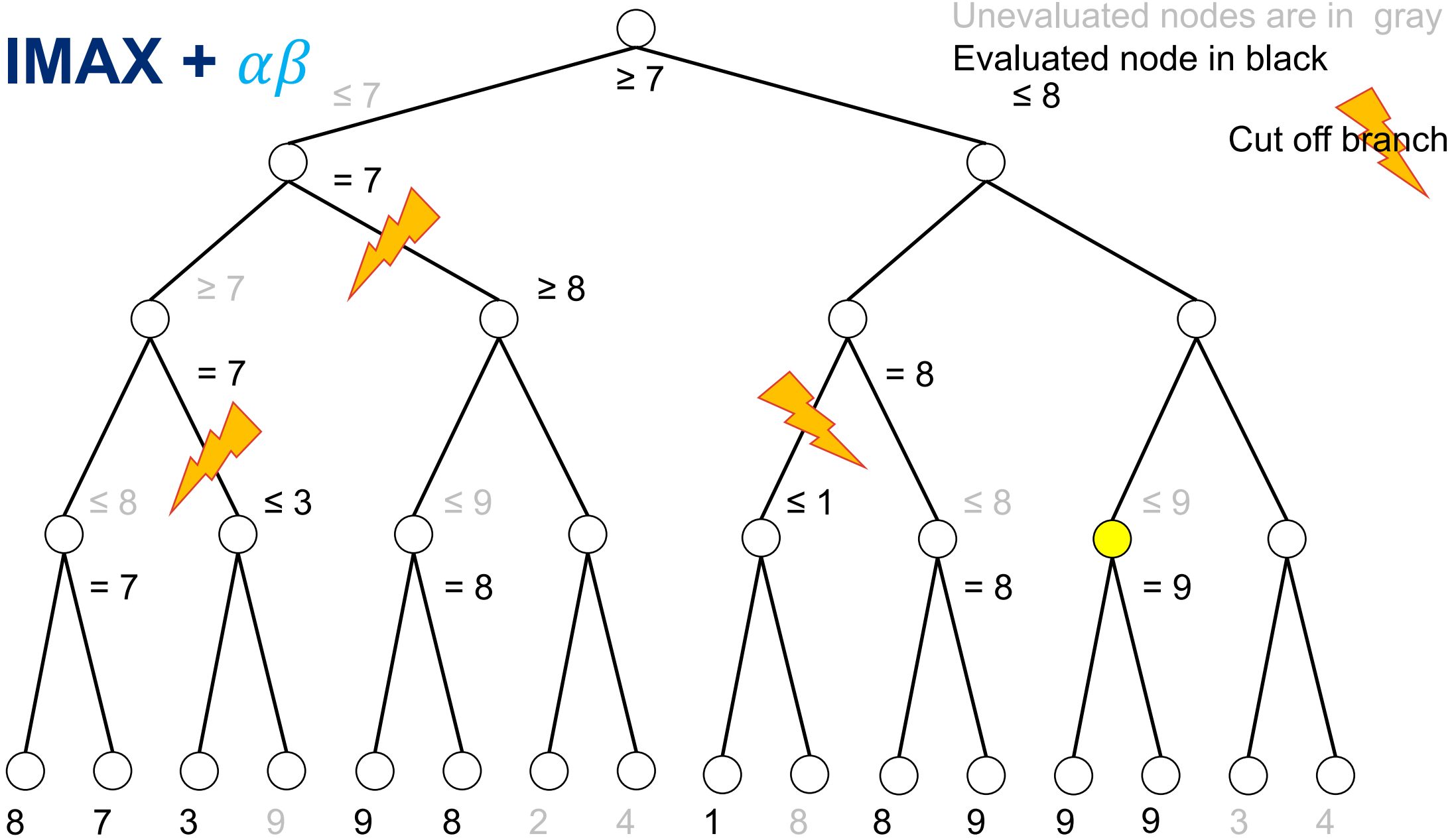
MINIMAX + $\alpha\beta$



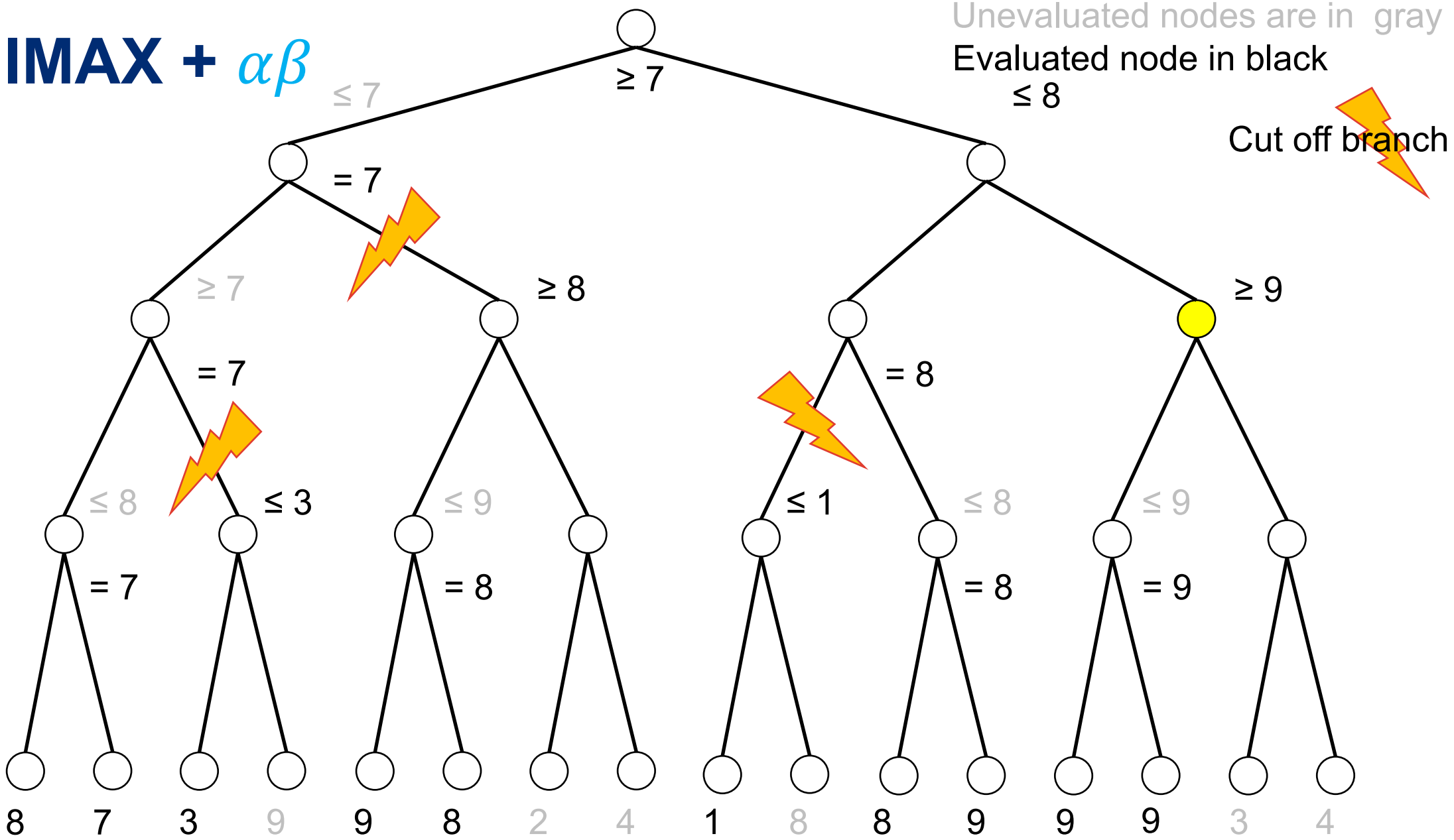
MINIMAX + $\alpha\beta$



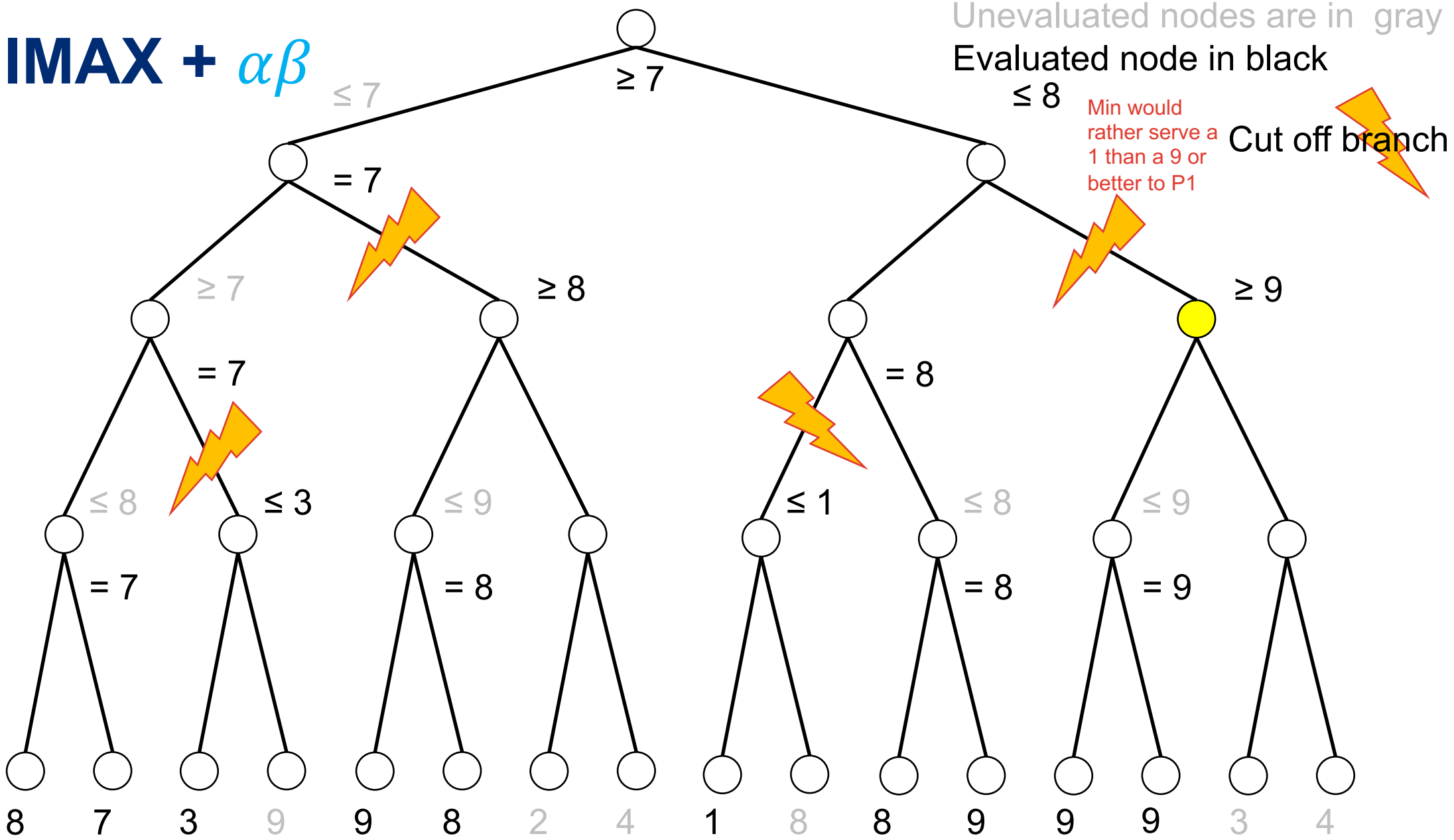
MINIMAX + $\alpha\beta$



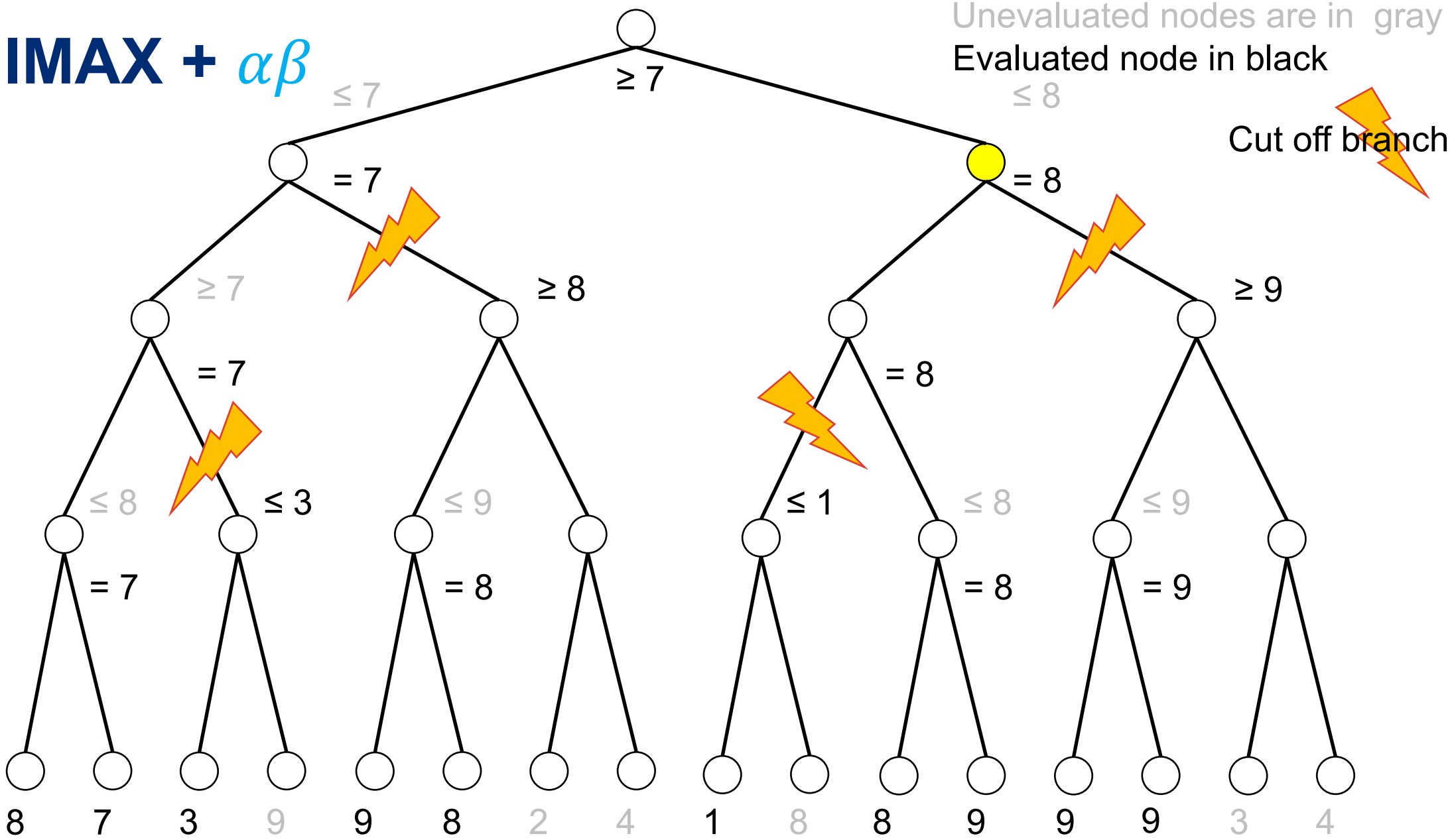
MINIMAX + $\alpha\beta$



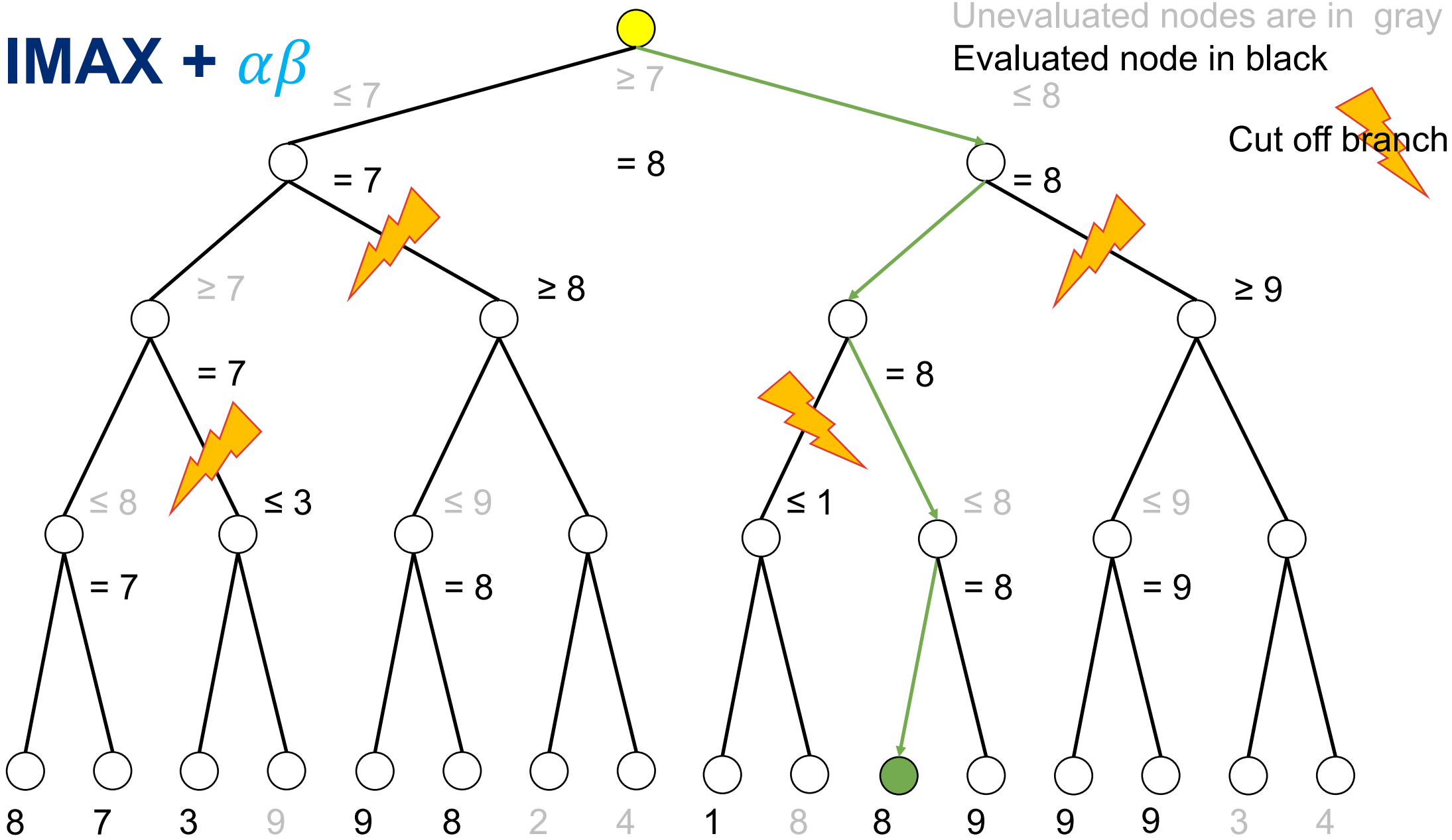
MINIMAX + $\alpha\beta$



MINIMAX + $\alpha\beta$



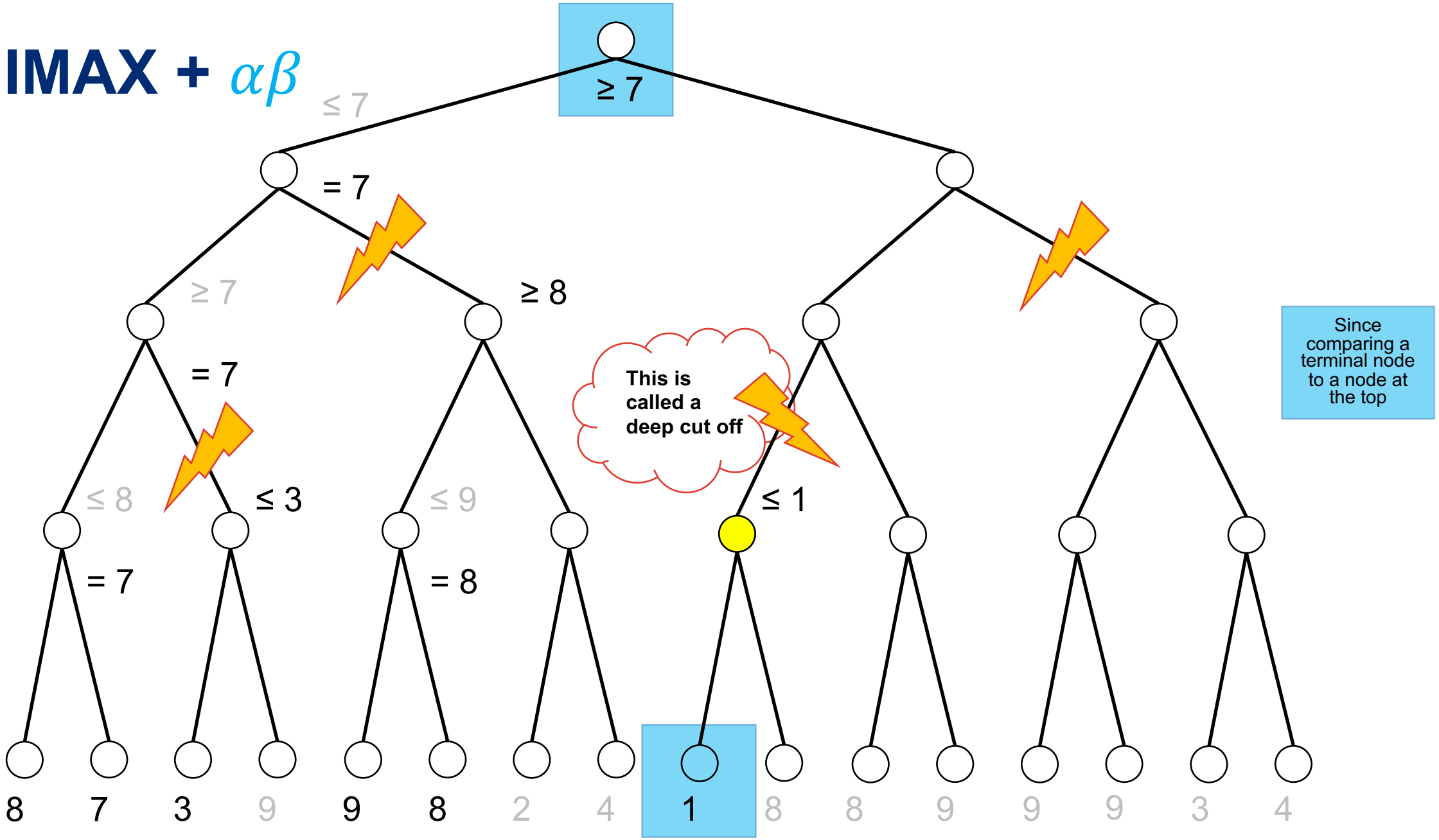
MINIMAX + $\alpha\beta$



MINIMAX + $\alpha\beta$

- $b^{d/2}$ calculations needed (on average)
- For the same resources we can go twice as deep
- Turns out looking ahead 7 levels is a mildly annoying computer to play against
- Looking ahead 15 levels is a computer that can beat a Grand Master (gross generalization, but this is what $\alpha\beta$ buys us)

MINIMAX + $\alpha\beta$



Since comparing a terminal node to a node at the top

JOHNS HOPKINS

APPLIED PHYSICS LABORATORY