

Ong Ming Sen
AI

ALPHA-BETA PRUNING FOR MINMAX ALGORITHM

Solving the Tic-Tac-Toe Game

Arif Nugraha Santosa

5025211048

Rayhan Arvianta Bayuputra

5025211217

Laurivasya Gadhind Syahafidh

5025211136

THE CONTENTS

- 1 What is the defenition of alpha-beta pruning?
- 2 What is Alpha ? What is Beta ?
- 3 What is the definition of minimax algorithm?
- 4 Improving Minimax algorithm with Alpha-beta pruning
- 5 Alpha beta pruning for minimax alogrithm in the tictactoe game
- 6 Alpha-Beta Pruning Implementation
- 7 Thank u!

WHAT IS THE DEFINITION OF ALPHA-BETA PRUNING?

Alpha-beta pruning adalah algoritma yang digunakan untuk mencari sebuah nilai maksimum dari suatu tree percobaan dengan bantuan limitasi nilai Alpha dan Beta nya untuk memutus rantai pencarian yang terlalu banyak.

WHAT IS ALPHA?

WHAT IS BETA?

Dalam konteks alpha-beta pruning, "alpha" dan "beta" adalah nilai yang digunakan sebagai **batas atas (alpha)** dan **batas bawah (beta)** untuk membatasi rentang penilaian yang relevan dalam pencarian algoritma.

Alpha (α) adalah nilai terbaik yang ditemukan untuk pemain maksimal (misalnya, komputer). Alpha mewakili nilai terbesar yang telah diketahui untuk pemain maksimal saat ini. Selama pencarian, jika ada nilai yang lebih besar dari alpha yang ditemukan, nilai alpha akan diperbarui.

Beta (β) adalah nilai terbaik yang ditemukan untuk pemain minimal (misalnya, lawan manusia). Beta mewakili nilai terkecil yang telah diketahui untuk pemain minimal saat ini. Selama pencarian, jika ada nilai yang lebih kecil dari beta yang ditemukan, nilai beta akan diperbarui.

WHAT IS THE DEFINITION OF MINIMAX ALGORITHM?

Algoritma Minimax adalah metode dalam kecerdasan buatan yang digunakan untuk memilih langkah terbaik dalam permainan. Dalam algoritma ini, pemain maksimal (komputer) berusaha untuk mendapatkan nilai tertinggi, sementara pemain minimal (manusia) berusaha untuk mendapatkan nilai terendah. Algoritma ini melakukan pencarian dalam tree permainan, mengevaluasi setiap langkah secara berulang hingga mencapai keadaan akhir permainan. Dengan cara ini, algoritma Minimax dapat menentukan langkah terbaik untuk pemain maksimal pada setiap tingkat kedalaman dalam permainan.

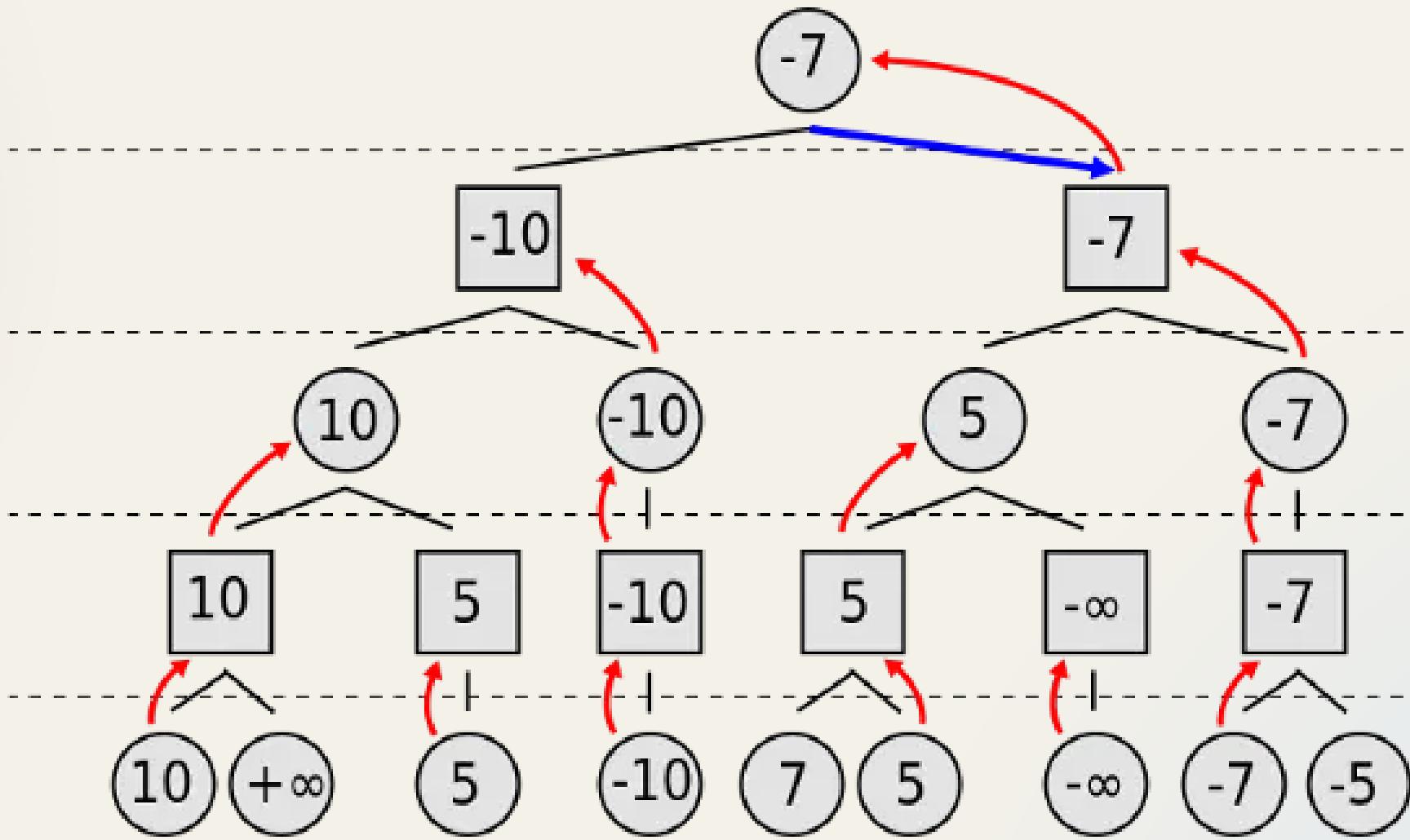
0 (max)

1 (min)

2 (max)

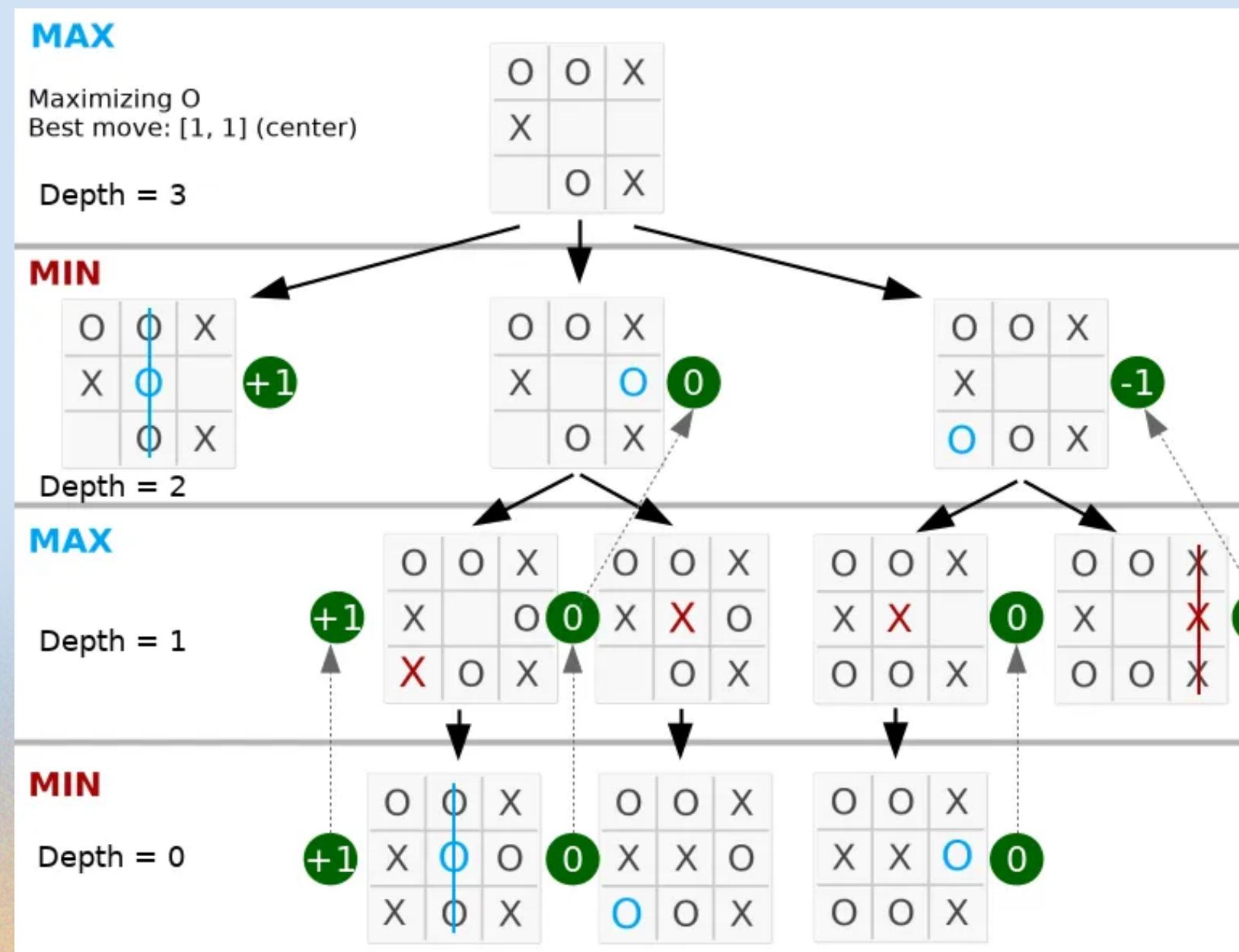
3 (min)

4 (max)



Algoritma ini akan bekerja dari bawah lalu ke atas membawa skor yang harus di pilih di setiap label nya, jika labelnya max artinya cabang tersebut harus memilih skor yang paling tinggi, jika labelnya min maka berlaku sebaliknya.

LALU, APA HUBUNGAN TICTACTOE DENGAN ALGORITMA MINMAX?



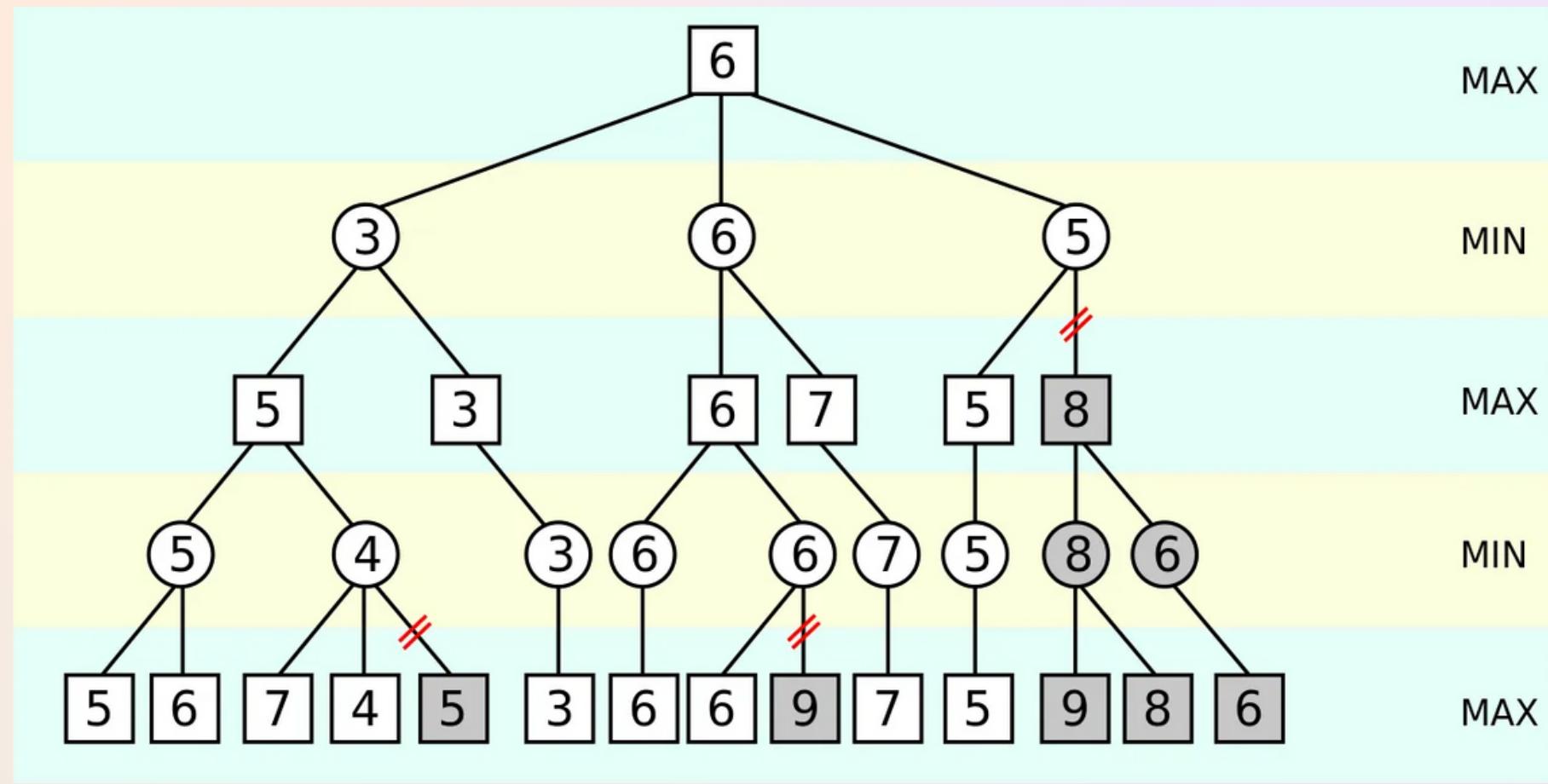
Fungsi dari pohon pencarian disamping adalah mensimulasikan setiap langkah yang bisa dilakukan pada permainan tic tac toe. Dengan:

- * 1 untuk Menang
- * 0 untuk draw / seri
- * -1 untuk kalah

Dengan cara ini, kita bisa mendapatkan skor mana yang paling tinggi dan dari node atau cabang mana skor optimal tersebut.

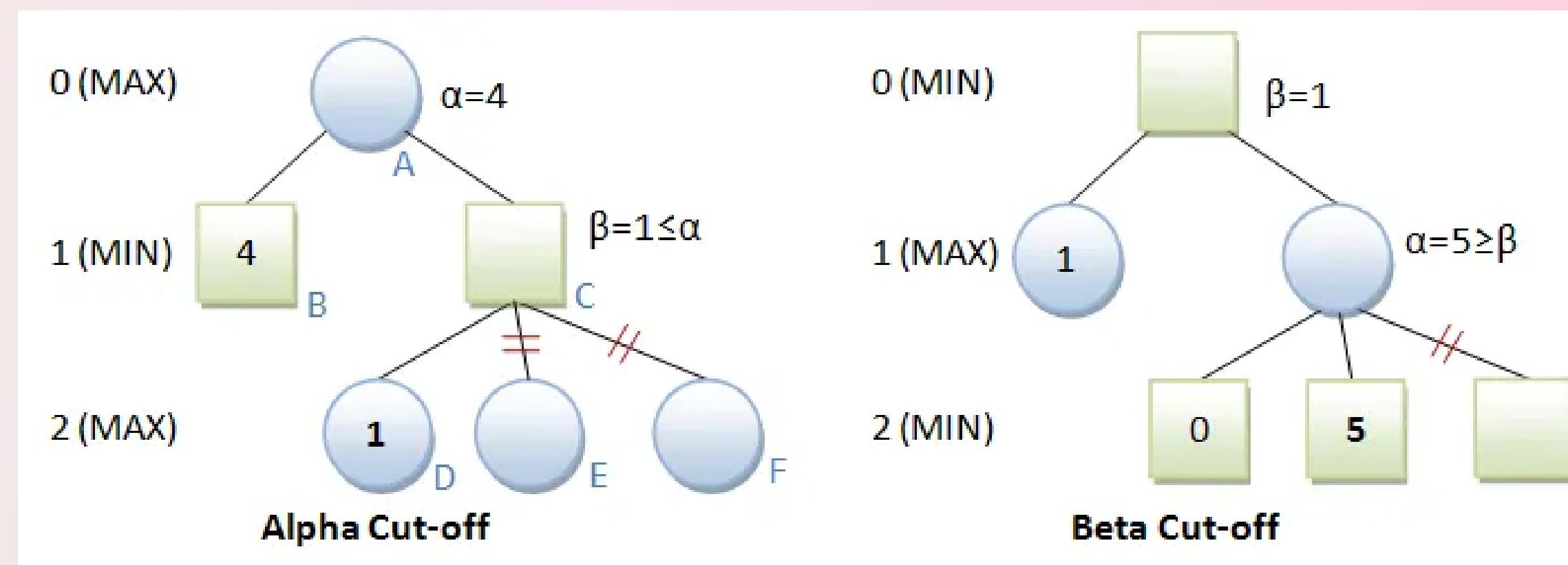
IMPROVING MINIMAX ALGORITHM WITH ALPHA-BETA PRUNING

Algoritma Minimax menghasilkan banyak node/cabang pencarian yang mana hal ini menyebabkan program akan banyak memakan sumber daya komputasi. Karena komputer akan terus membuat simulasi disetiap cabang sampai permainan berakhir di sisi cabang – cabangnya. Oleh karena itu, kita akan menyempurnakannya menggunakan algoritma Alpha-Beta pruning, yaitu dengan memangkas node dari pohon cabang jika sudah ada dibawah nilai Alpha dan Beta.

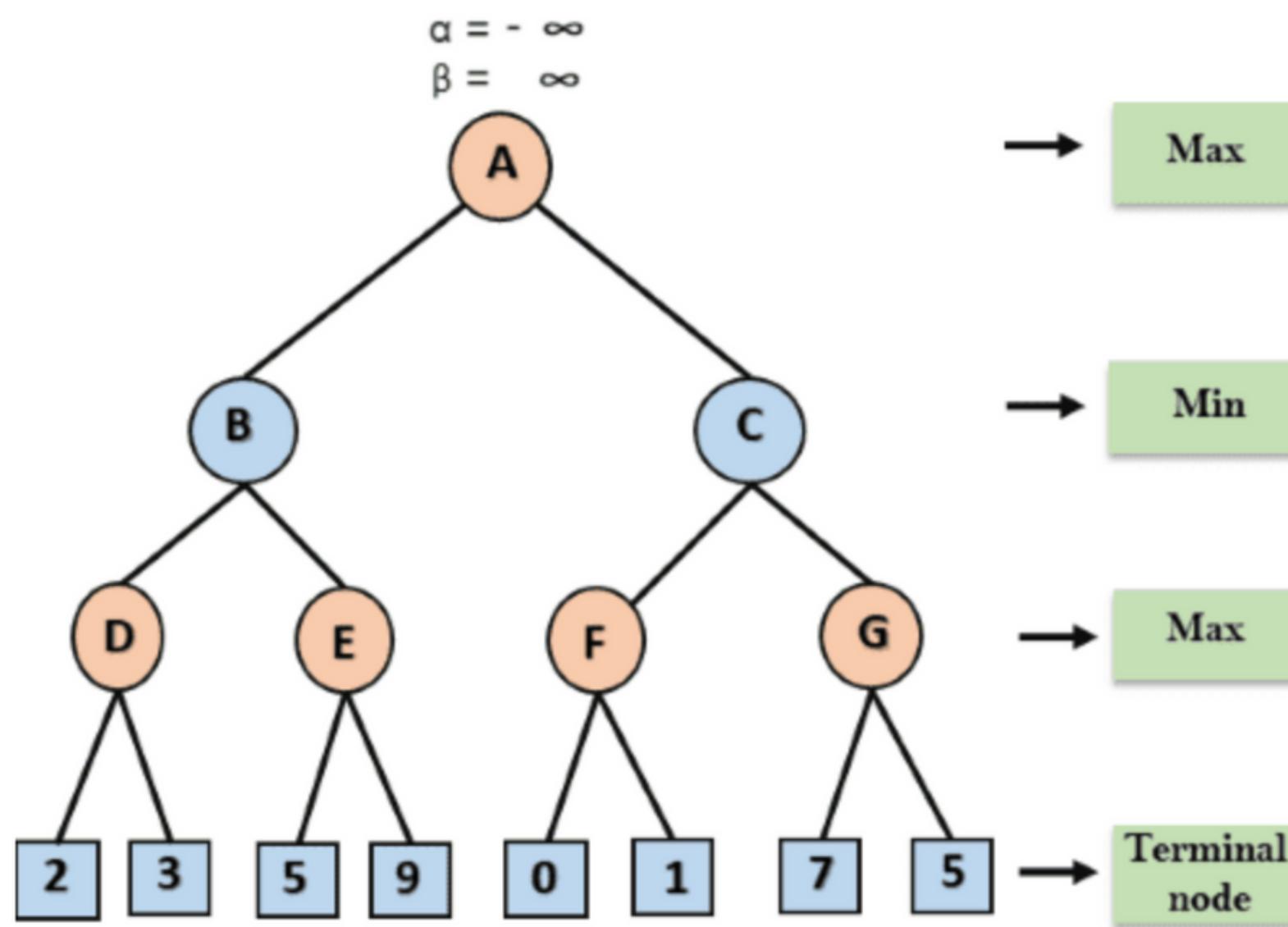


pada ilustrasi disamping nilai awal dari Alpha adalah Negatif tak hingga, dan nilai Beta adalah Positif tak hingga. Nilai ini akan di perbarui terus selama pencarian cabang (minimizing / maximizing).

Lalu pencarian cabang akan di hentikan ketika **nilai Alpha nya lebih besar atau sama dengan nilai Beta**. Yang artinya nilai yang semula Beta > Alpha, ini akan berhenti ketika nilai nya berubah menjadi Alpha > Beta.



CONTOH PENYELESAIAN ALPHA-BETA PRUNING

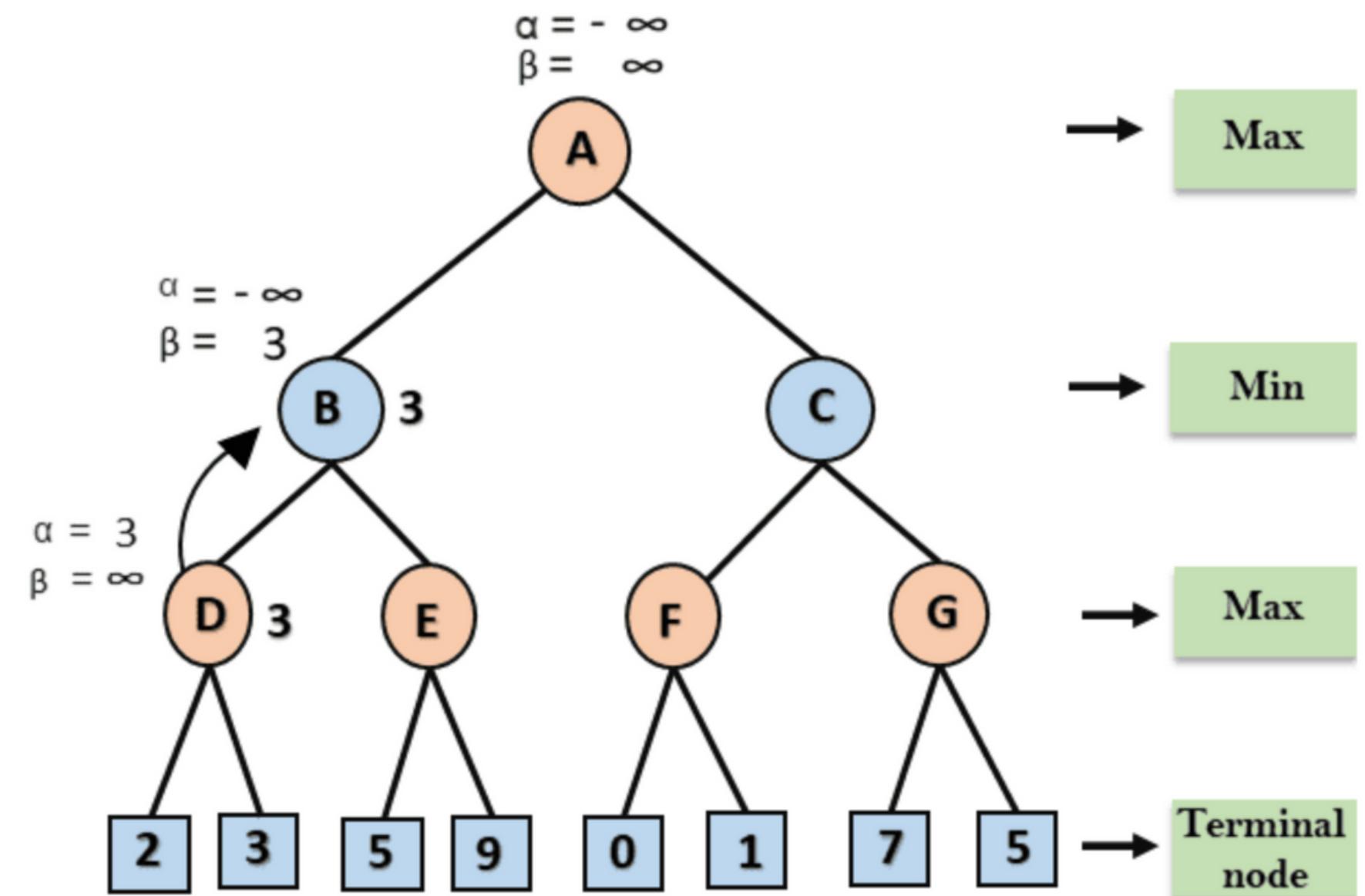


Langkah 1:

Pemain Max akan mulai dengan bergerak dari simpul A, di mana $\alpha = -$ dan $\beta = +$, dan meneruskan nilai-nilai ini alpha dan beta ke simpul B, di mana sekali lagi $\alpha = -$ dan $\beta = +$, dan Simpul B meneruskan nilai yang sama ke anaknya D

Langkah 2:

Nilai α akan ditentukan sebagai giliran Max di Simpul D. Nilai α akan dibandingkan dengan 2, kemudian 3, dan nilai α di simpul D akan menjadi maksimum antara $(2, 3) = 3$, dan nilai simpul juga akan menjadi 3.

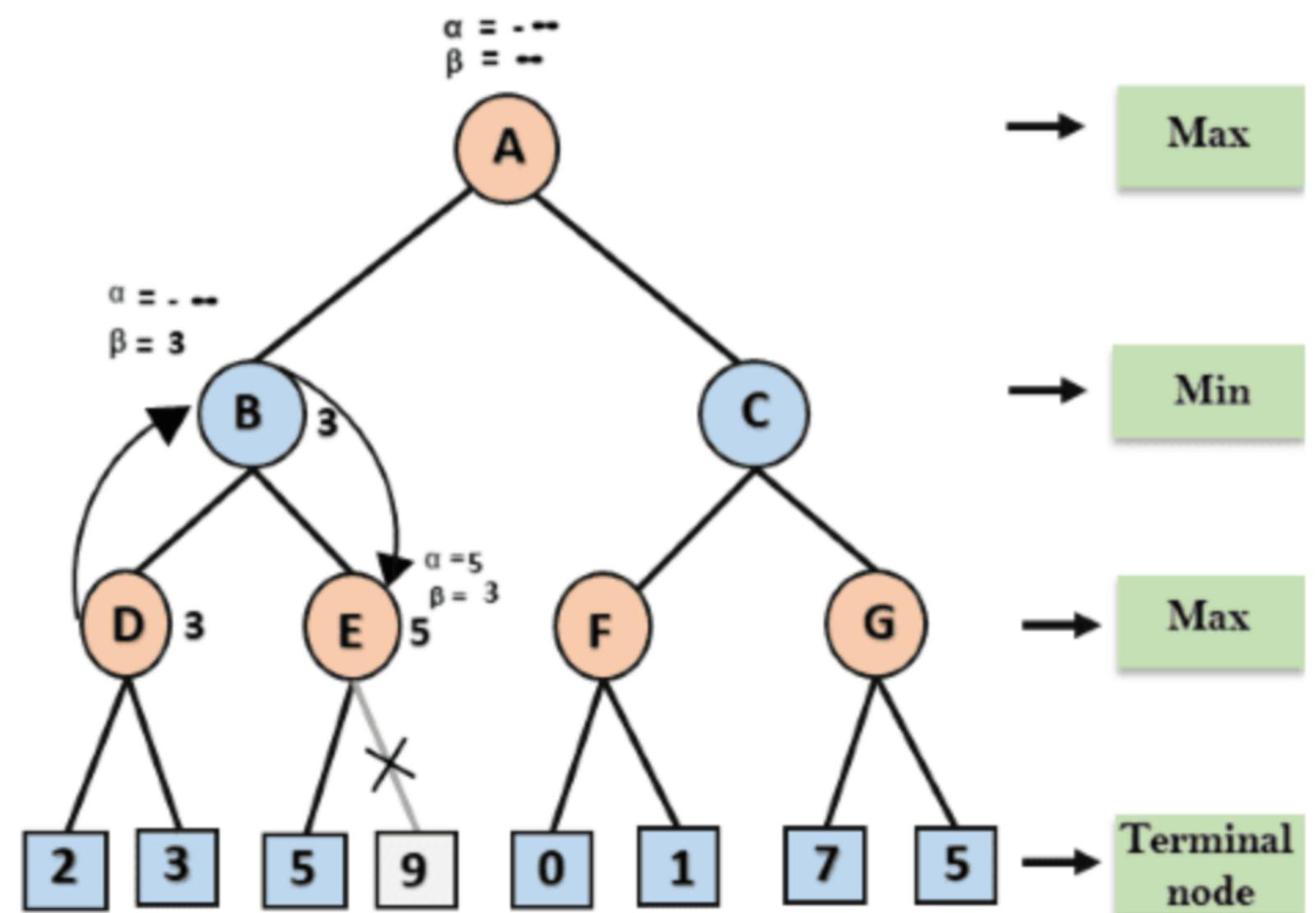


Langkah 3:

Algoritma sekarang kembali ke simpul B, di mana nilai α akan berubah karena ini adalah giliran Min, sekarang $\alpha = +$, dan akan dibandingkan dengan nilai simpul berikutnya yang tersedia, yaitu $\min(\alpha, 3) = 3$, jadi pada simpul B sekarang $\alpha = -$, dan $\beta = 3$.

Langkah 4:

Max akan mengambil giliran di simpul E, mengubah nilai alpha. Nilai alpha saat ini akan dibandingkan dengan 5, menghasilkan $\max(-, 5) = 5$, dan pada simpul E $\alpha = 5$ dan $\beta = 3$, di mana $\alpha \geq \beta$, penerus kanan E akan dipangkas, dan algoritma tidak akan melintasinya, dan nilai pada simpul E akan menjadi 5.

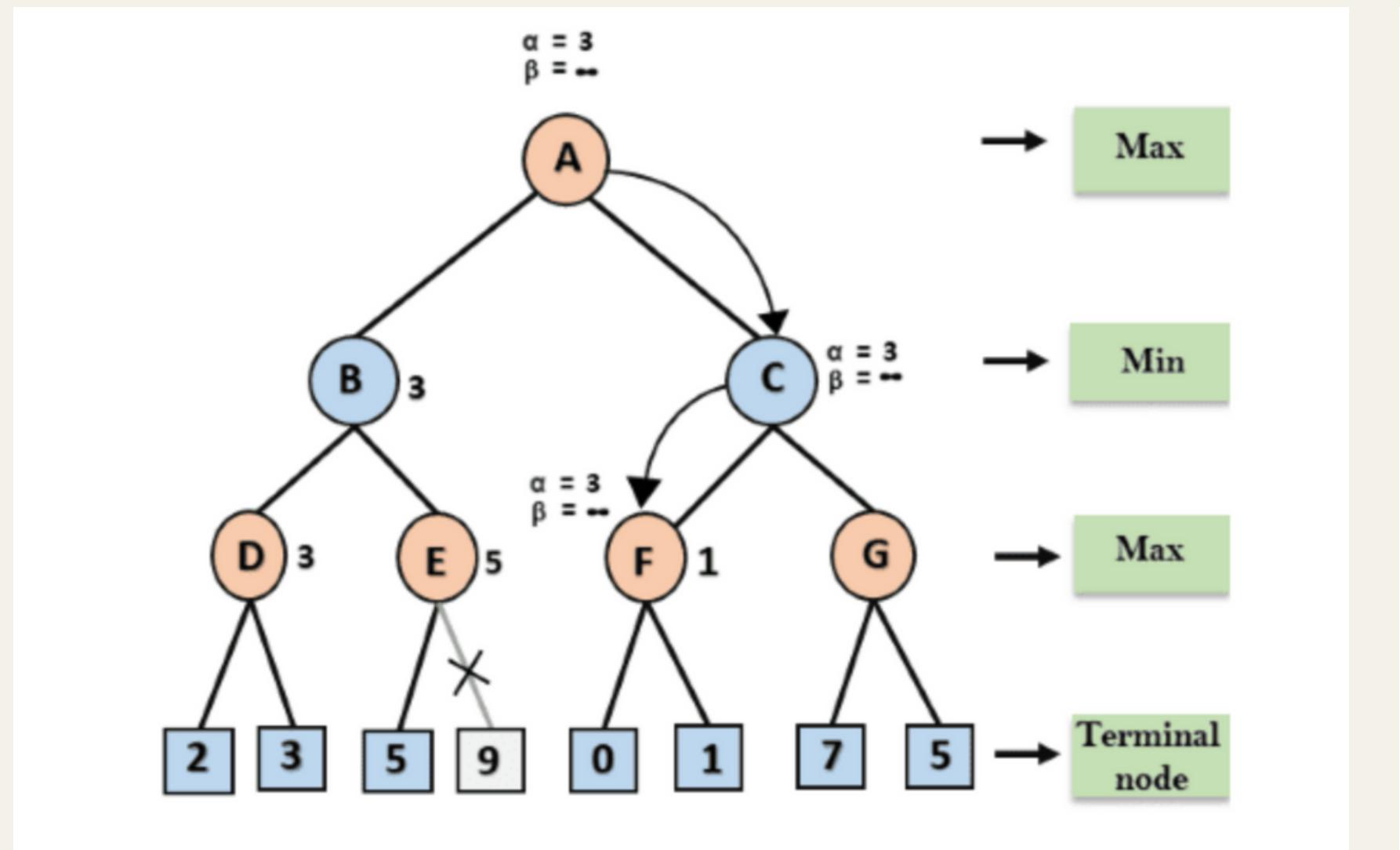


Langkah 5:

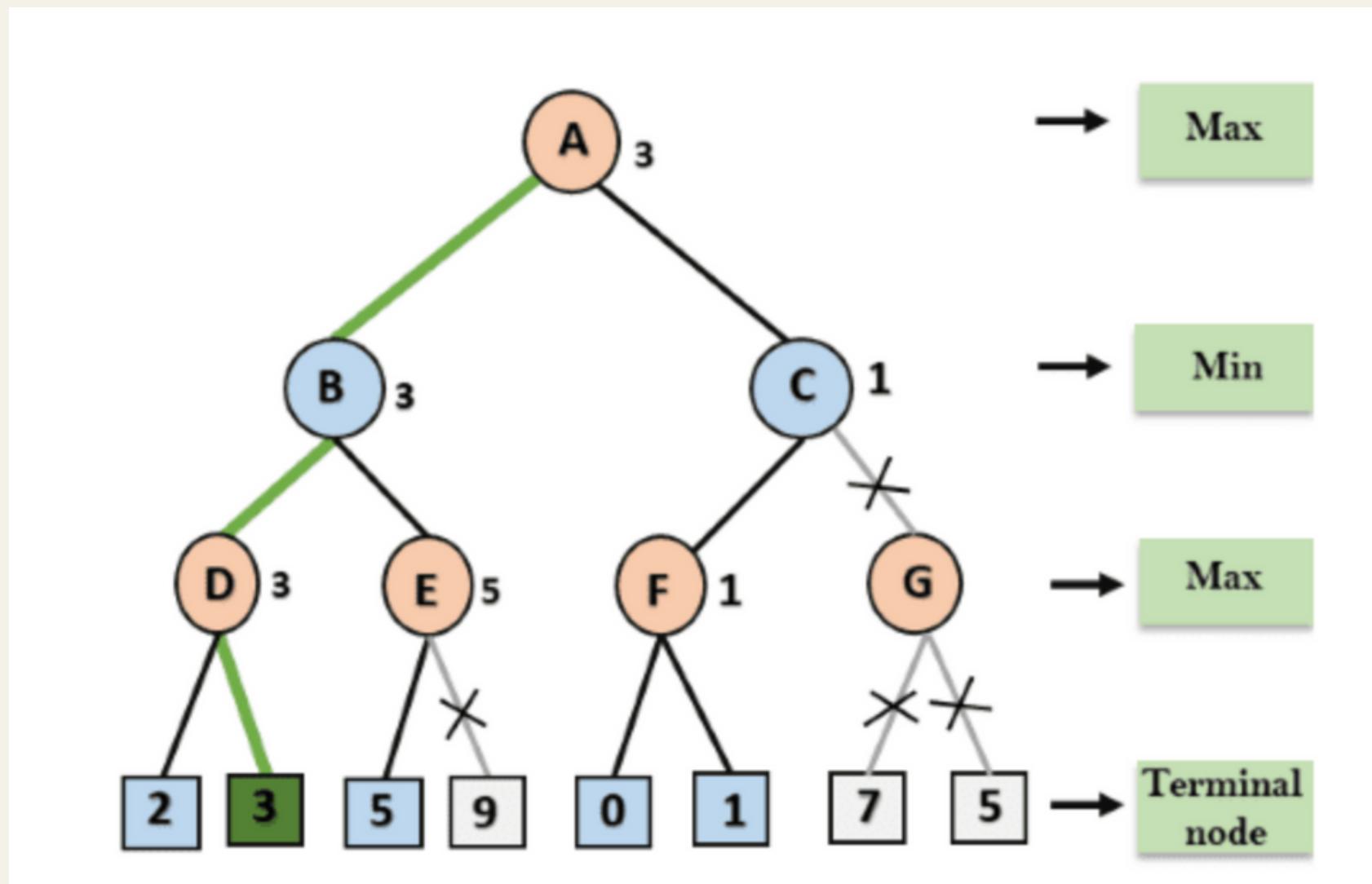
Metode ini sekarang mundur ke pohon, dari simpul B ke simpul A. Nilai alpha akan dimodifikasi di simpul A, dan nilai tertinggi yang tersedia adalah 3 sebagai max (-, 3) = 3, dan $\alpha = +$. Dua nilai ini akan ditransfer ke penerus kanan A, simpul C. $\alpha = 3$ dan $\beta = +$ akan dilewatkan ke simpul F di simpul C, dan nilai yang sama akan dilewatkan ke simpul F.

Langkah 6:

Nilai α akan dibandingkan dengan anak kiri, yaitu 0, dan $\max(3, 0) = 3$, dan kemudian dengan anak kanan, yaitu 1, dan $\max(3, 1) = 3$ akan tetap sama, tetapi nilai simpul F akan berubah menjadi 1.



Langkah 7:
 Simpul F mengembalikan nilai simpul 1 ke simpul C, pada C $\alpha = 3$ dan $\beta = +$, nilai beta dimodifikasi, dan dibandingkan dengan 1, menghasilkan $\min(\alpha, 1) = 1$. Sekarang, pada C, $\alpha = 3$ dan $\beta = 1$, dan sekali lagi, memenuhi kondisi $\alpha \geq \beta$, algoritma akan memangkas penerus berikutnya dari C, yaitu G, dan tidak akan menghitung seluruh sub-pohon G.



Langkah 8:
 Sekarang C mengembalikan 1 ke A, dengan $\max(3, 1) = 3$ menjadi hasil terbesar untuk A. Pohon permainan lengkap, yang menunjukkan simpul yang dihitung dan belum dihitung, ditunjukkan di bawah ini. Sebagai hasilnya, dalam kasus ini, nilai maksimum terbaik adalah 3.

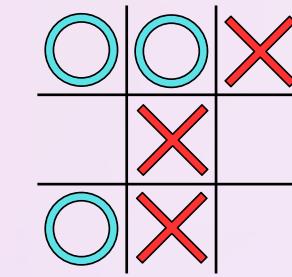
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME

$\alpha = -\infty$

$\beta = +\infty$

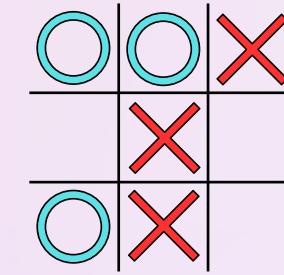
$\alpha \geq \beta$? False

score = $-\infty$

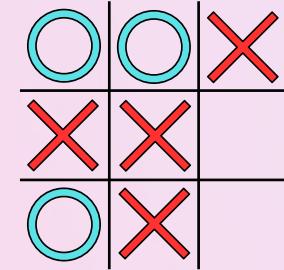


ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME

$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $-\infty$

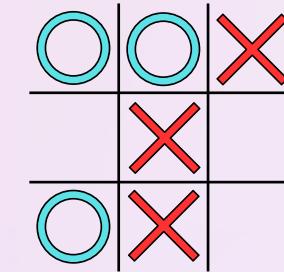


$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $+\infty$

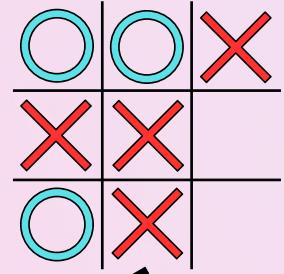


ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME

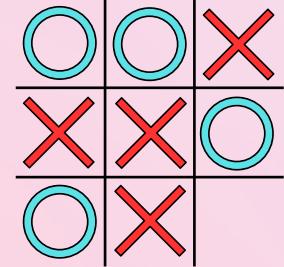
$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $-\infty$



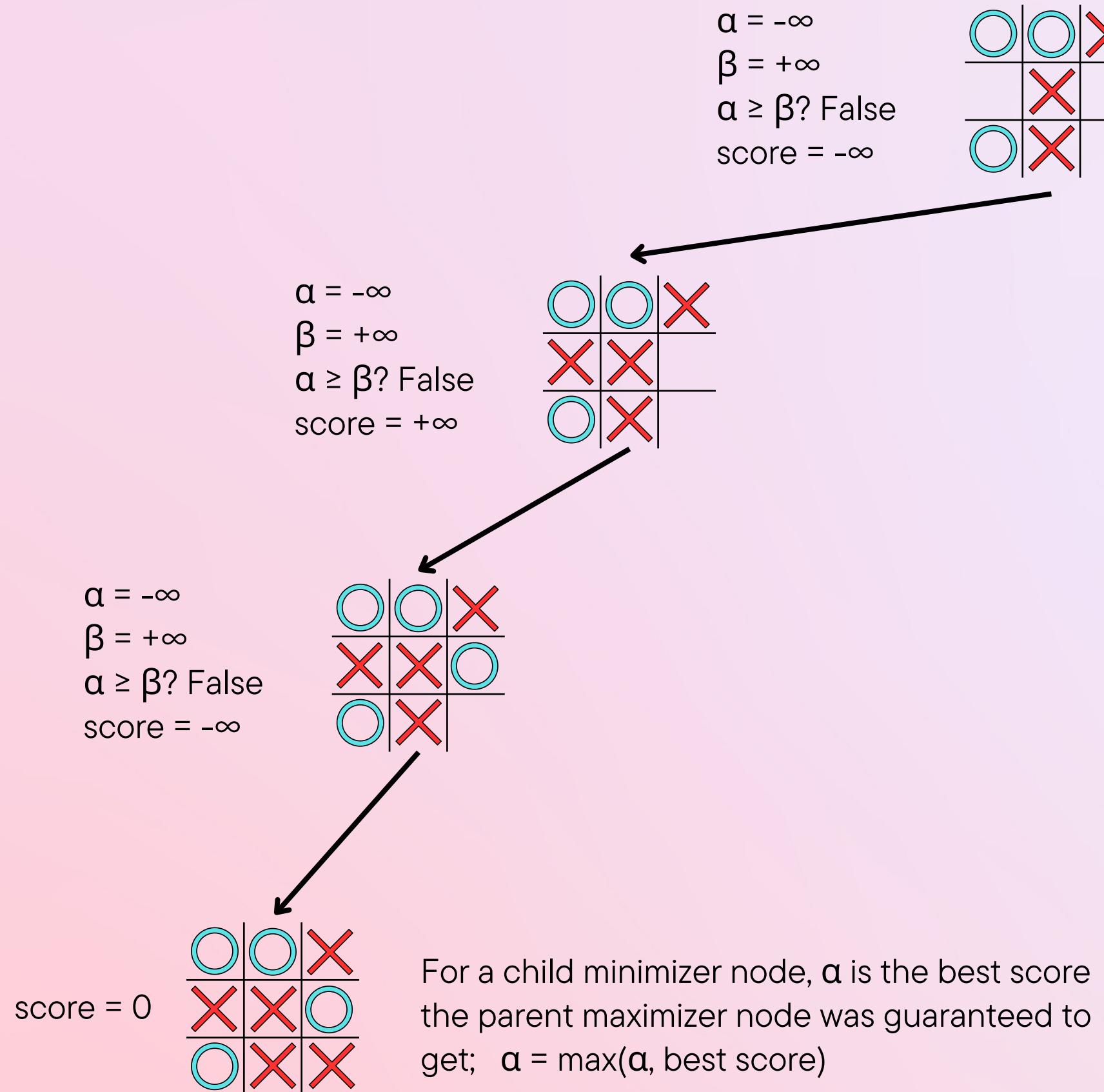
$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $+\infty$



$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $-\infty$

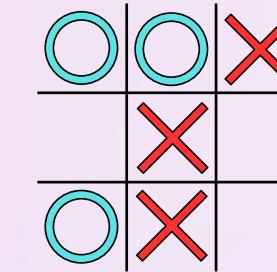


ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME

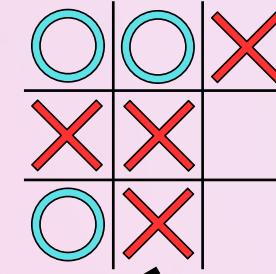


ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME

$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $-\infty$

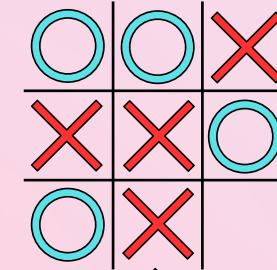


$\alpha = -\infty$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = $+\infty$

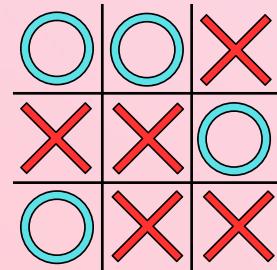


For a child maximizer node, β is the best score
the parent minimizer node was guaranteed to
get; $\beta = \min(\beta, \text{best score})$

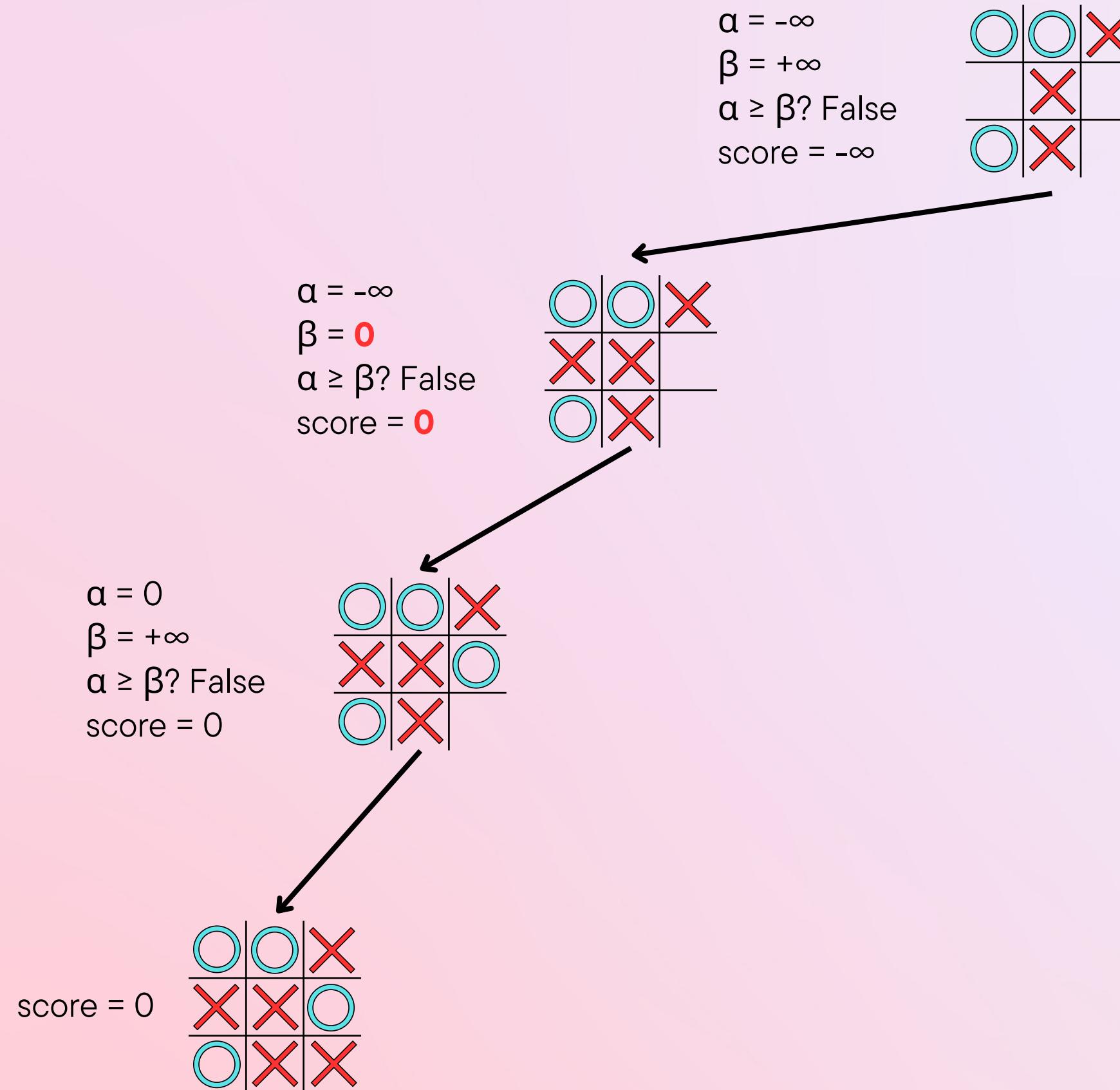
$\alpha = 0$
 $\beta = +\infty$
 $\alpha \geq \beta?$ False
score = 0



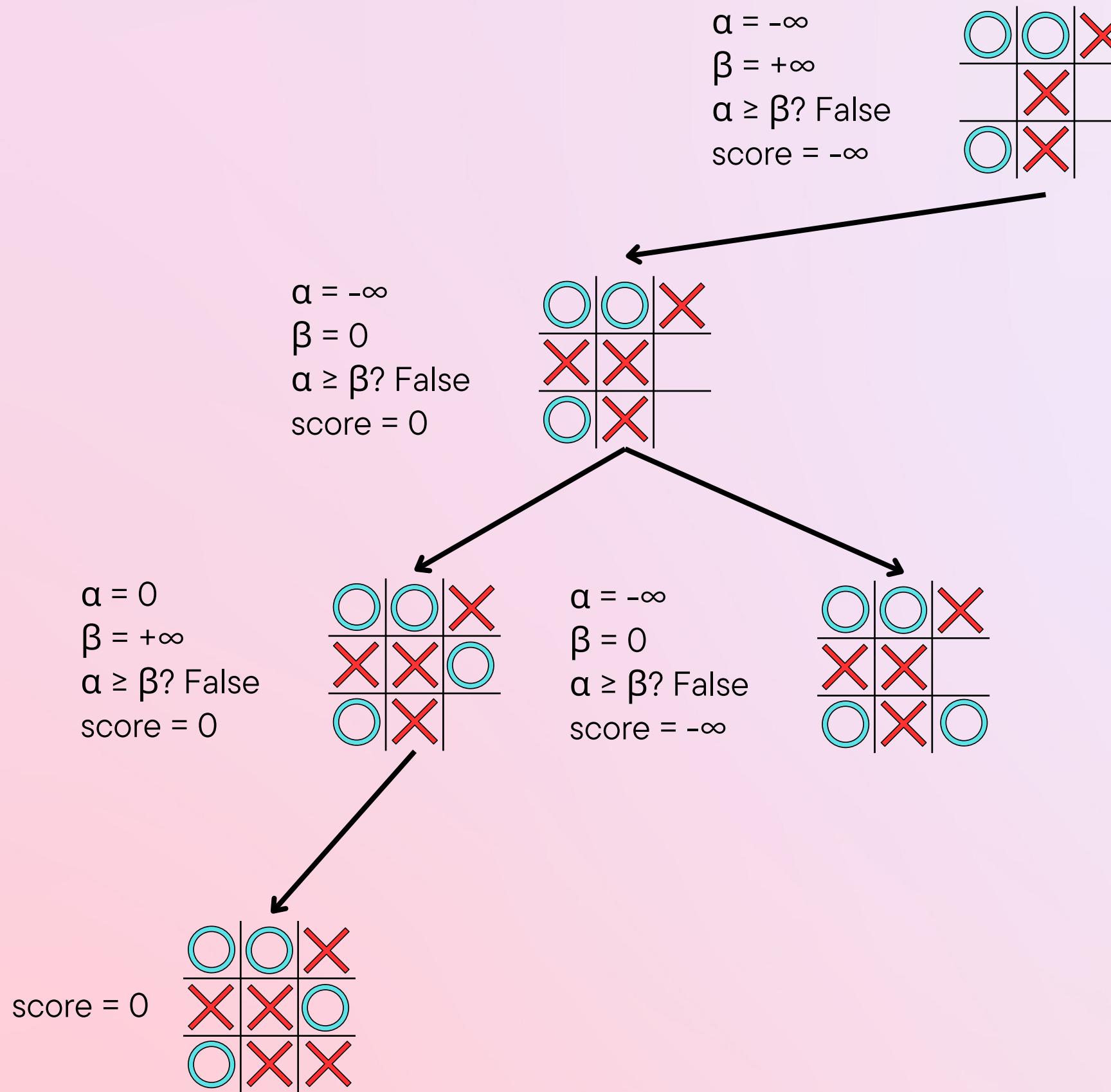
score = 0



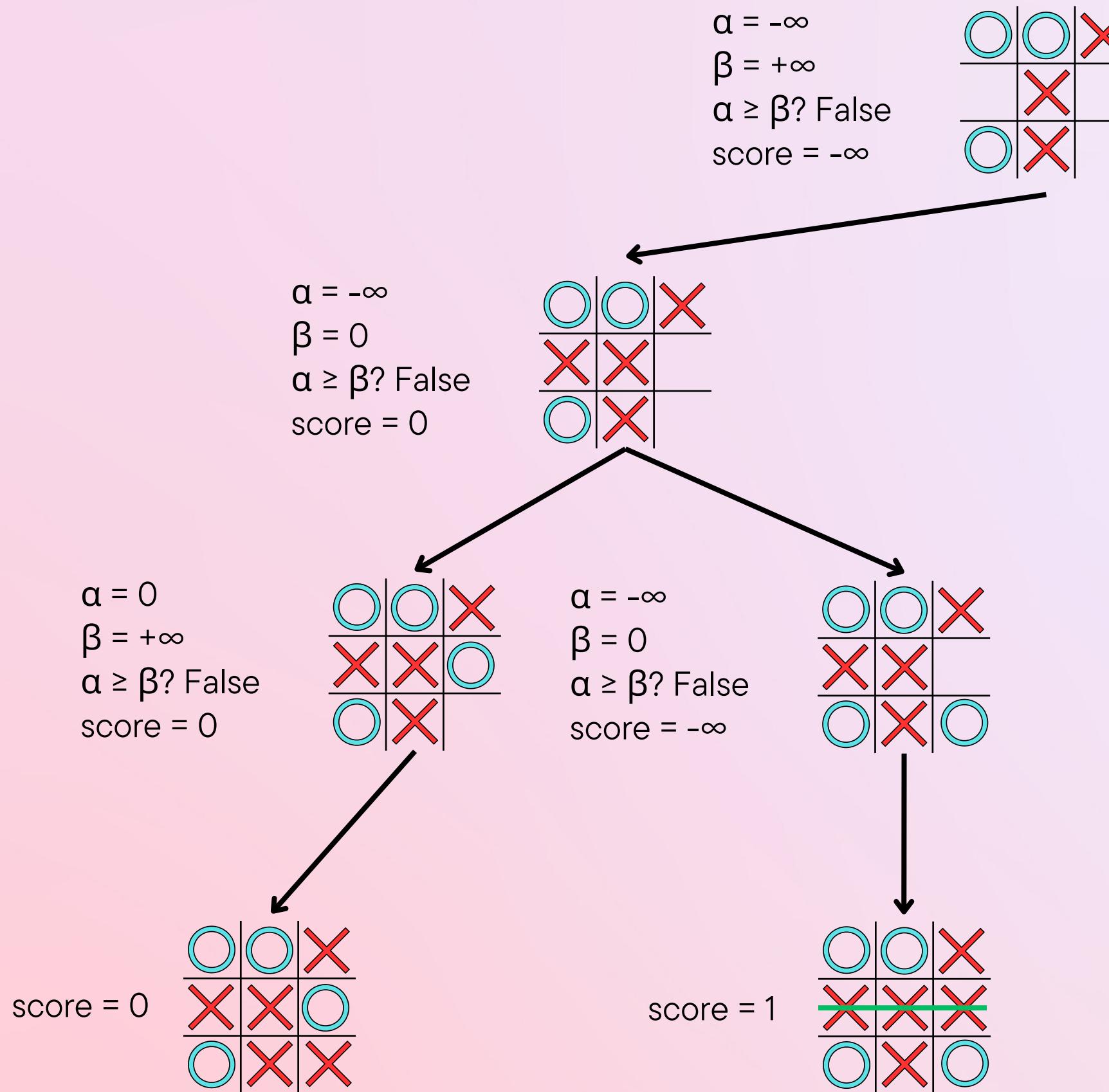
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



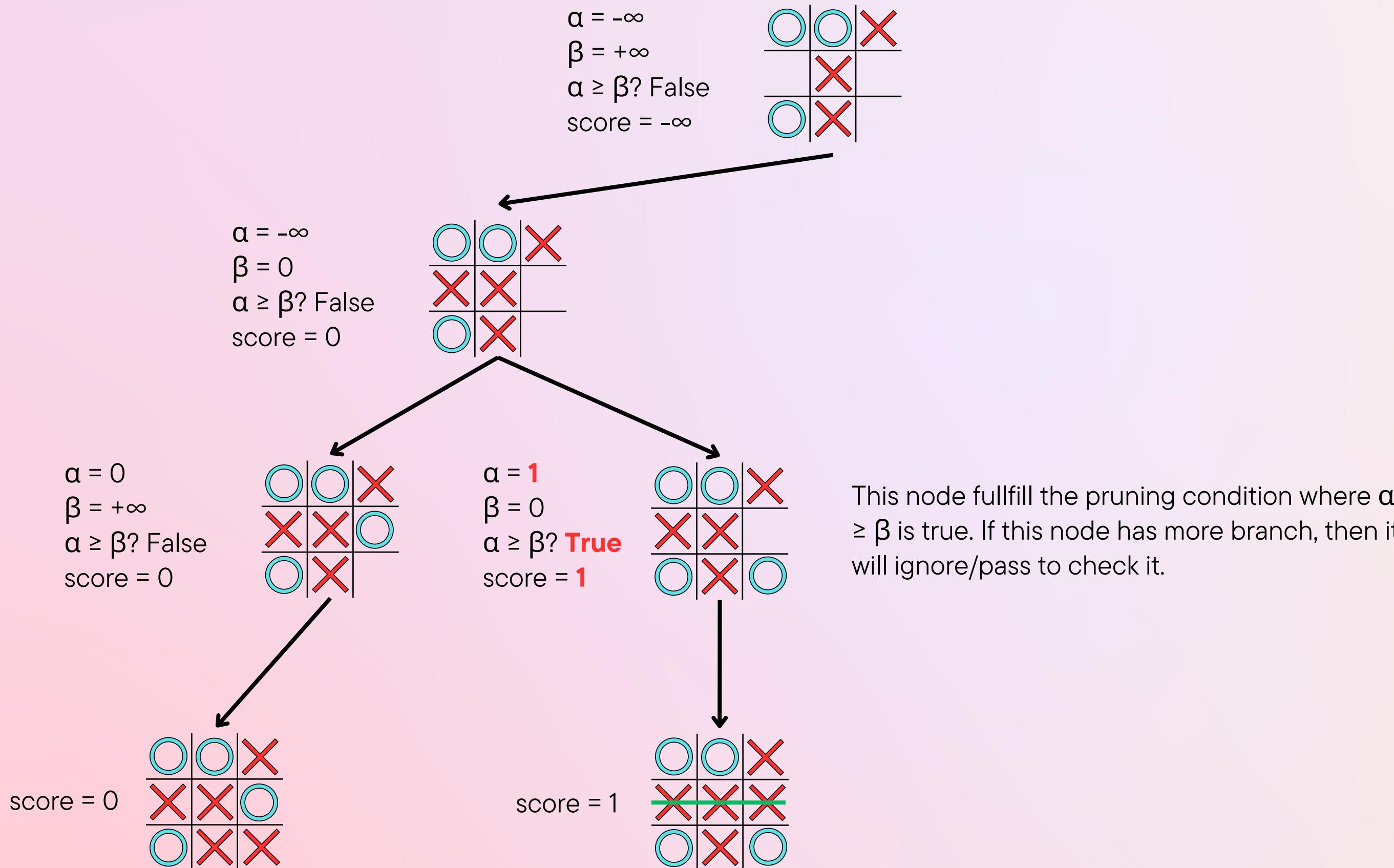
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



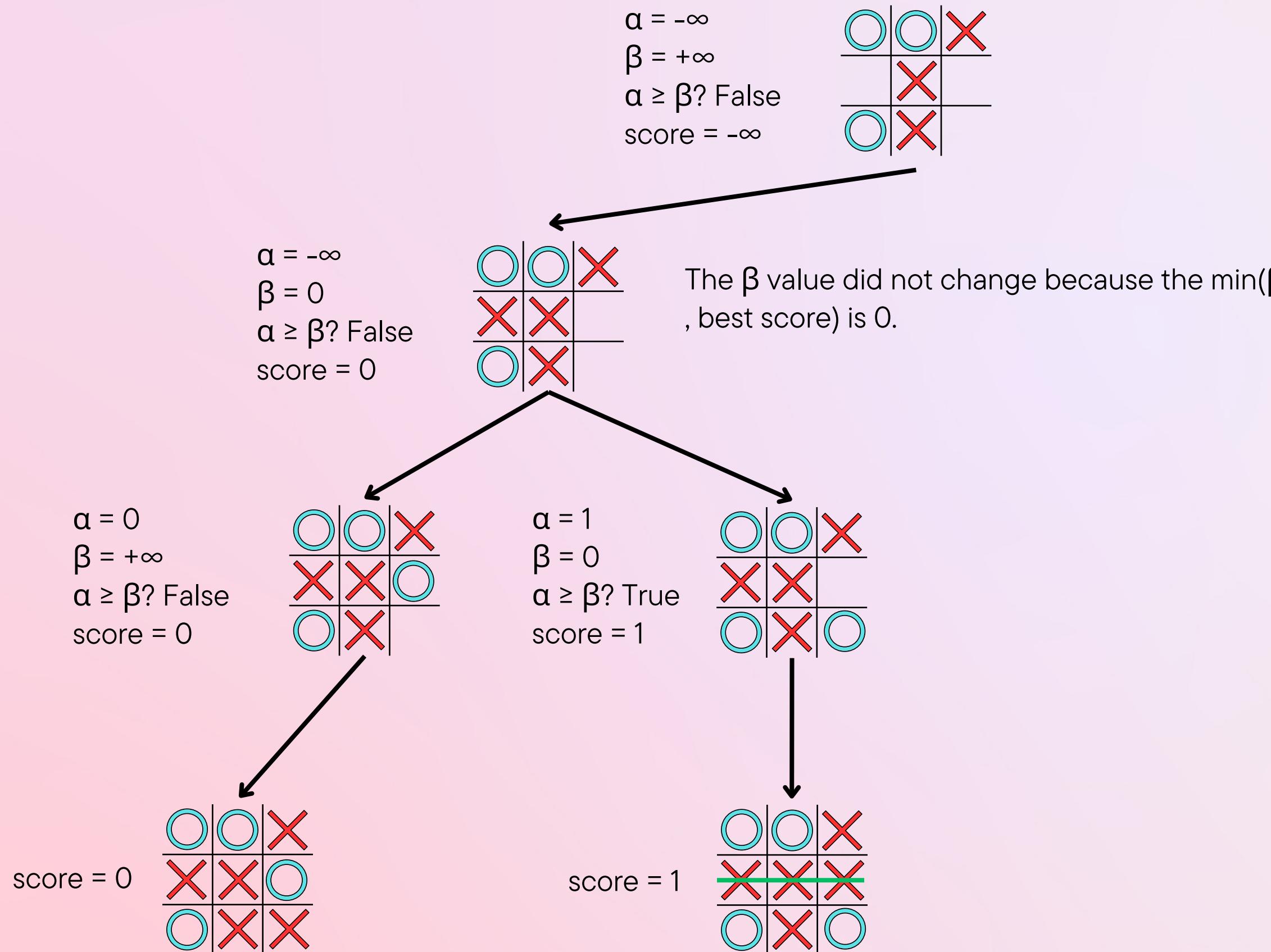
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



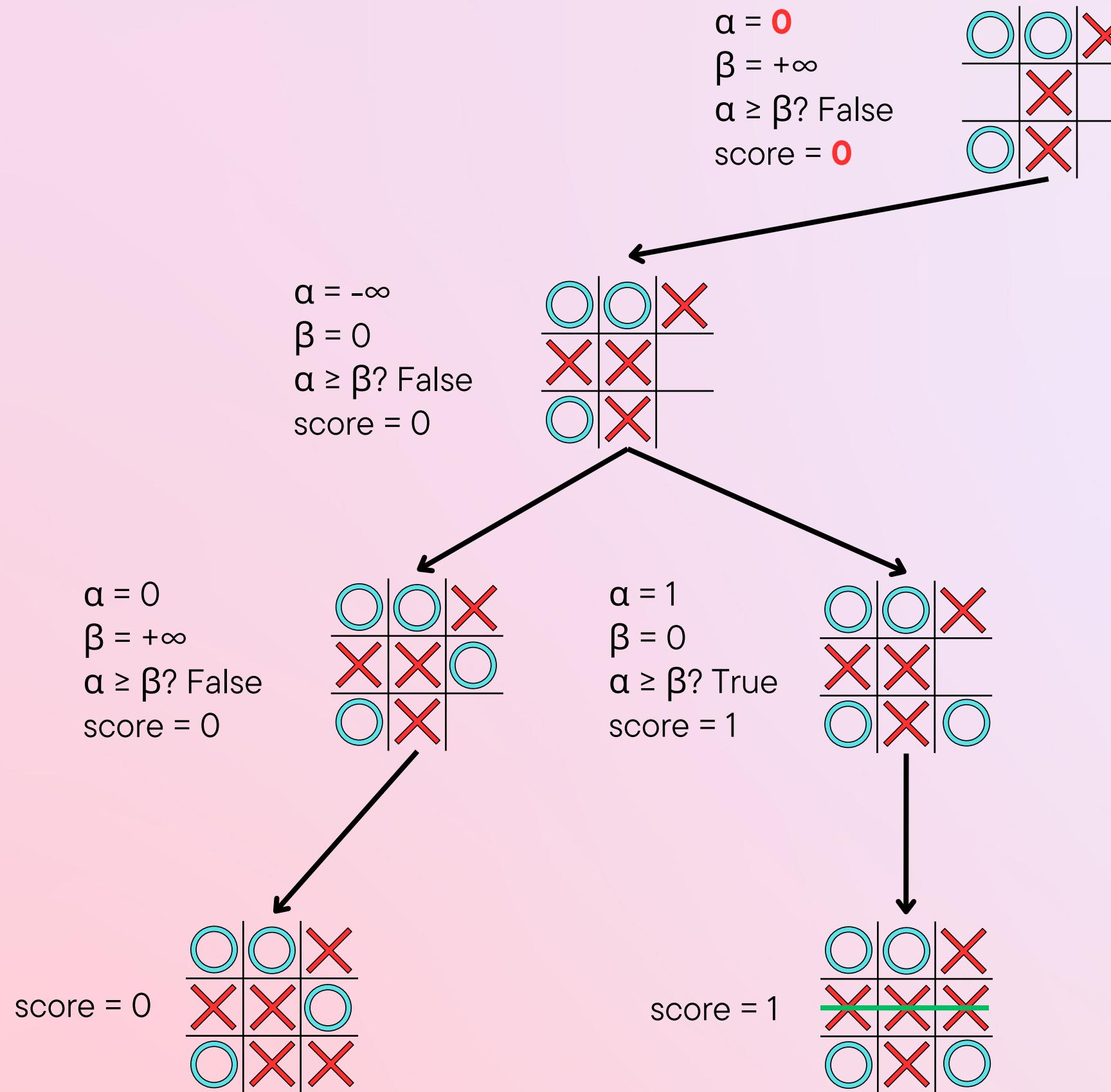
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



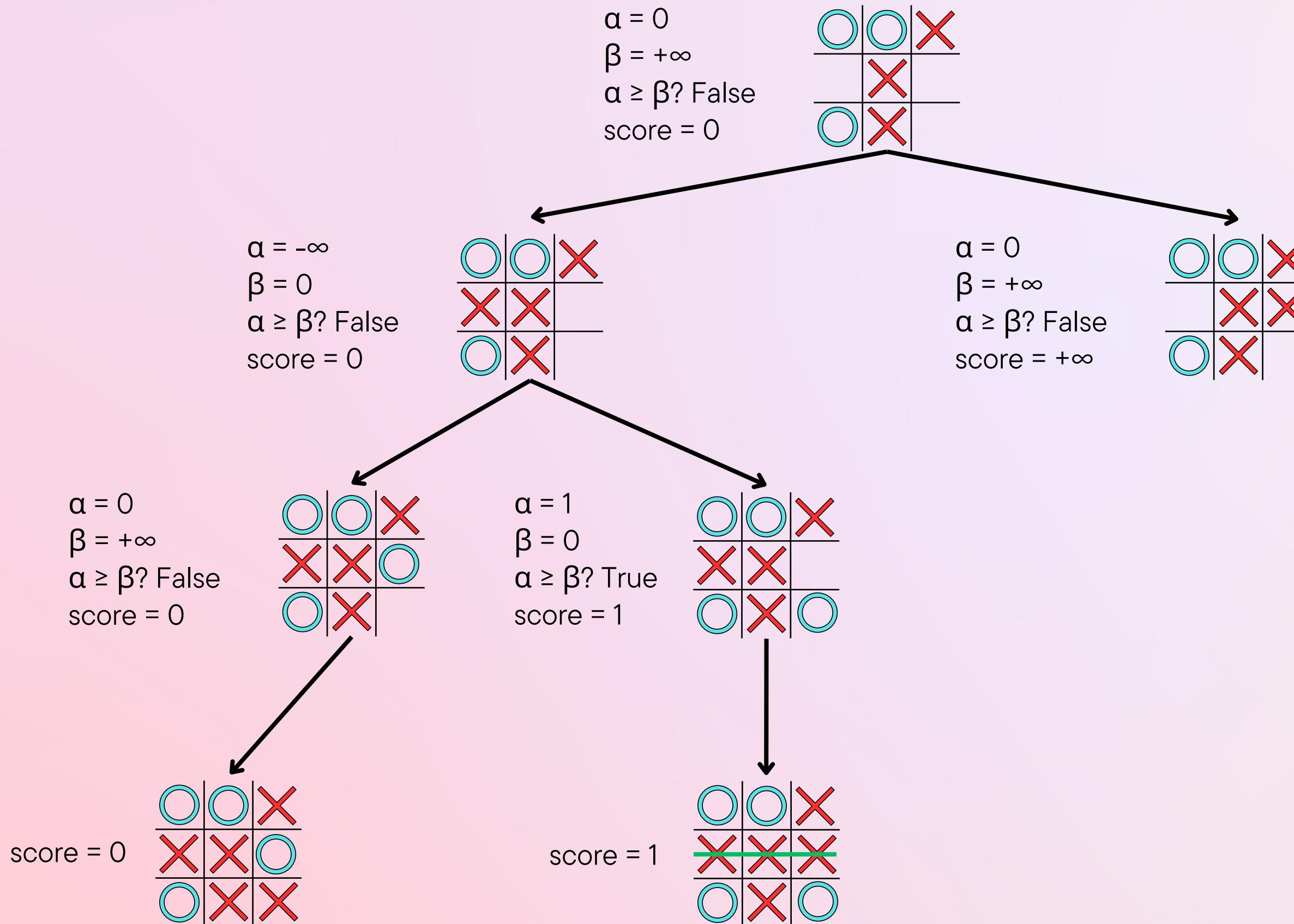
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



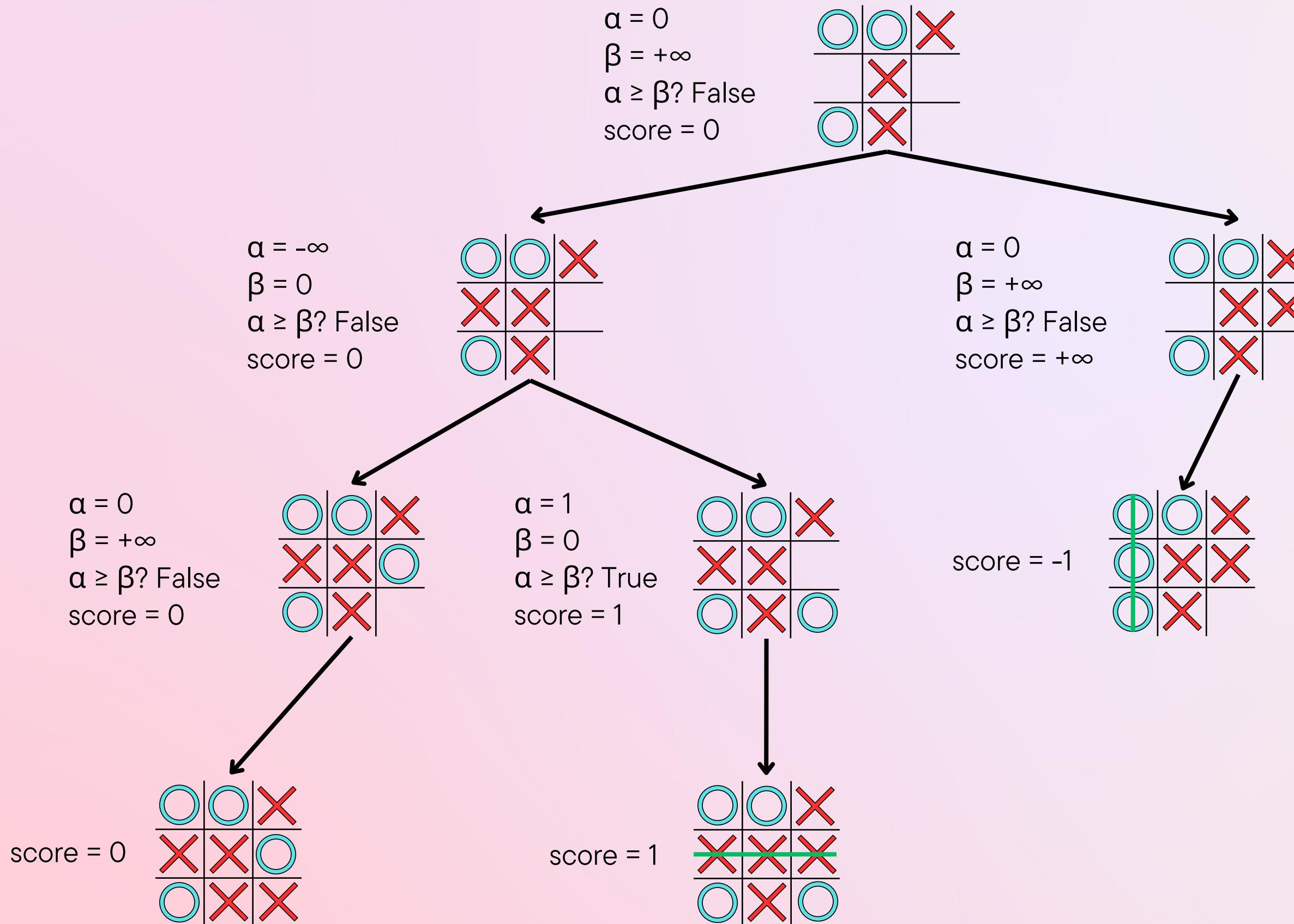
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



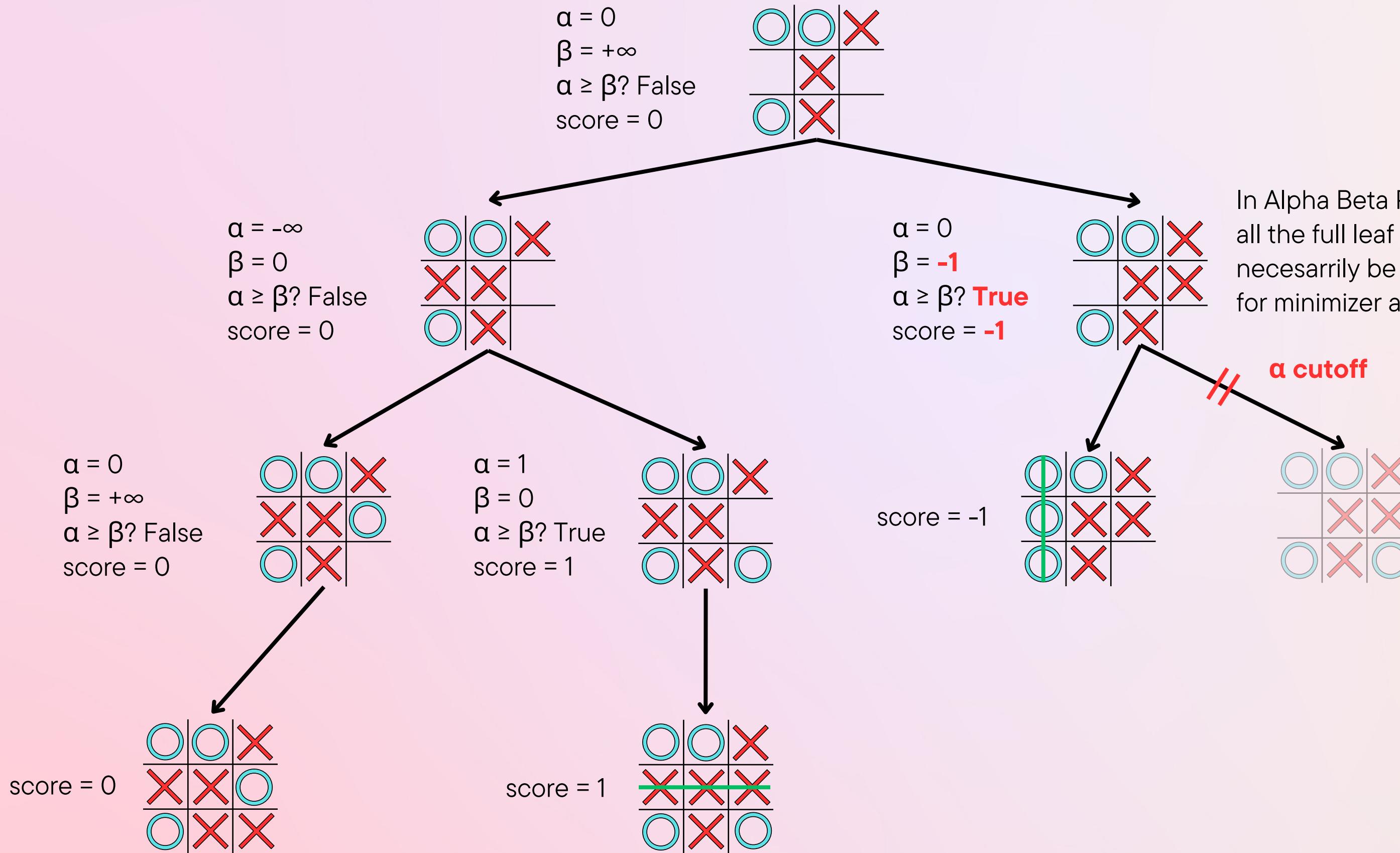
ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



ALPHA BETA PRUNING FOR MINIMAX ALGORITHM IN THE TICTACTOE GAME



THANK U FOR
LISTENING ^^

OMSA TEAM 2023