Game

Shinta Oktaviana Mkom

Shinta.oktaviana@tik.pnj.ac.id

Tujuan Perkuliahan

- Mahasiswa memahami definisi game dan jenis-jenis games
- Mahasiswa memahami algoritma minimax dan alfa-beta Pruning
- Mahasiswa memahami contoh-contoh dari implementasi algoritma tersebut

Pendahuluan

- Dimulai tahun 1950
- Game pertama : catur oleh Konard Zuse,Claude Shannon, Norbert Wiener, dan Alan Turing
- Lebih real dari pada problem searching (uninform maupun Inform search)
- Prunning digunakan agar hasil cepat tercapai
- Fungsi heuristic digunakan agar searching dilakukan hanya kepada node yang mengarah ke hasil (efisiensi memory searching)

Tipe-tipe game

- Game dengan informasi lengkap

  Pemain mengetahui semua langkah yang mungkin terjadi dari dirinya sendiri dan dari lawan dan hasil akhir dari permaian mereka. Contoh tic tac toe dan catur

- Game dengan informasi tak lengkap

  Pemain tidak mengetahui semua kemungkinan langkah lawan. Contoh permaian kartur poker ato bridge.

- Deterministik

  Perilaku game yang mudah ditentukan, karena tidak mengandung unsur ketidakpastian.

- Non Deterministik

  Perilaku game yang tidak mudah ditentukan (kebalikan dari deterministik), contohnya permaian taktik pertempuran.

Defined Game Playing

- Initial state : posisi awal termasuk pihak mana yang memberikan aksi
- Set of operator : kumpulan aturan yang menjadi batasan bagi pemain
- Terminal test : posisi kapan permainan berakhir
- Utility function/payoff function : angka yang dihasilkan ketika permainan selesai

Game Playing : Catur

- Initial state : posisi awal termasuk pihak mana yang memberikan aksi
- Set of operator : aturan-aturan pergerakan catur. Misal pion hanya boleh bergerak maju satu langkah, benteng bergerak vertikal dan horizontal, dll
- Terminal test : salah satu raja di makan pihak lawan
- Utility function/payoff function : 1 untuk pihak yang menang, -1 untuk pihak yang kalah, 0 jika kondisi seri.

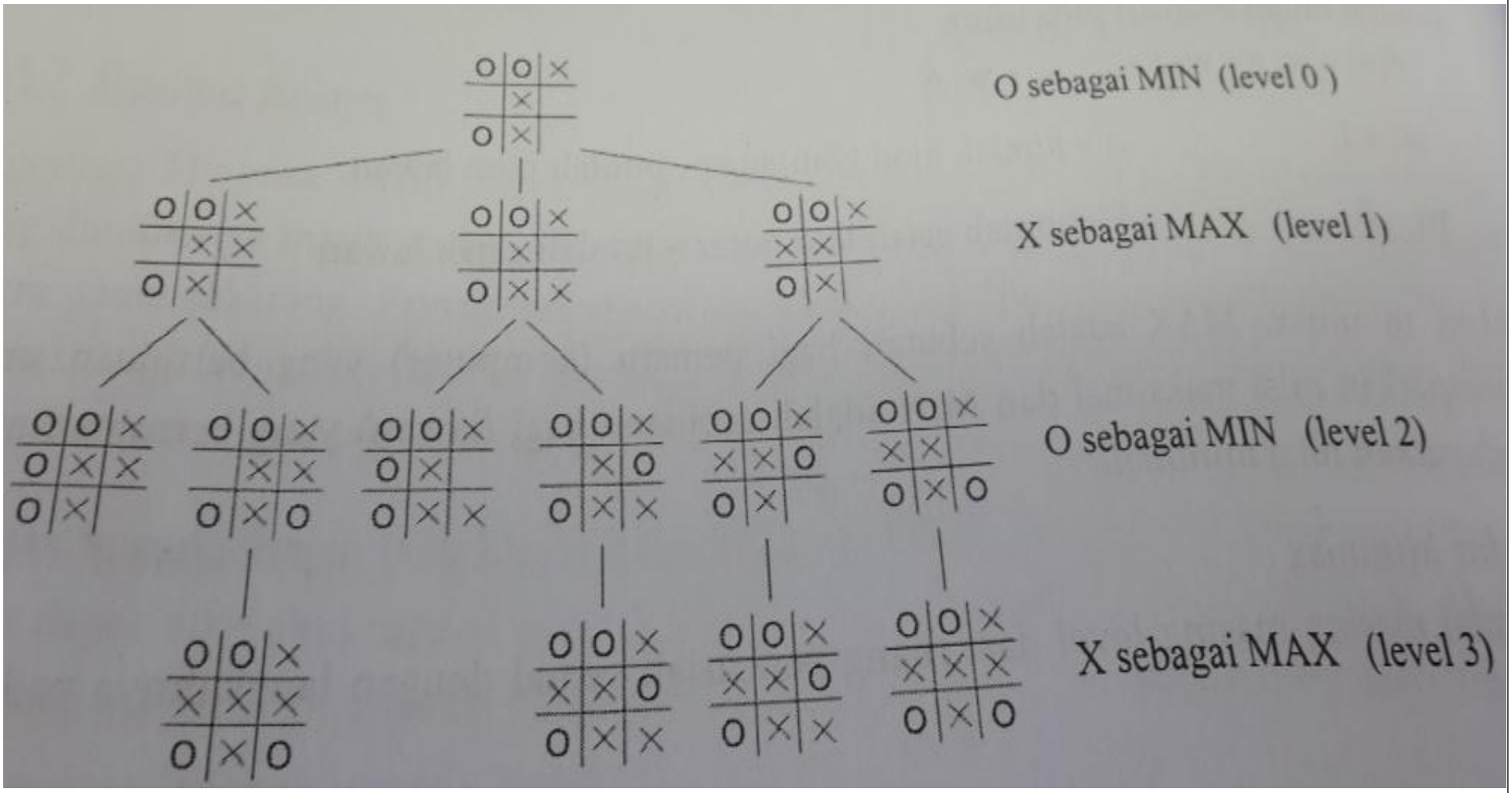
Algorithm - MiniMax

- *Dikembangkan pada tahun 1928 oleh John Von Neuman*
- *Minimax merupakan algoritma AI yang sering digunakan untuk membuat AI jenis decision (game untuk 2 pemain). Tugas dari minimax adalah memetakan seluruh kemungkinan permainan dimana masing-masing hasil akhir permainan punya nilai Heuristik. Tujuan dari minimax adalah memilih peluang terkecil agent kalah atau memilih peluang terbesar agar agent bisa menang.*

Tahapan MiniMax

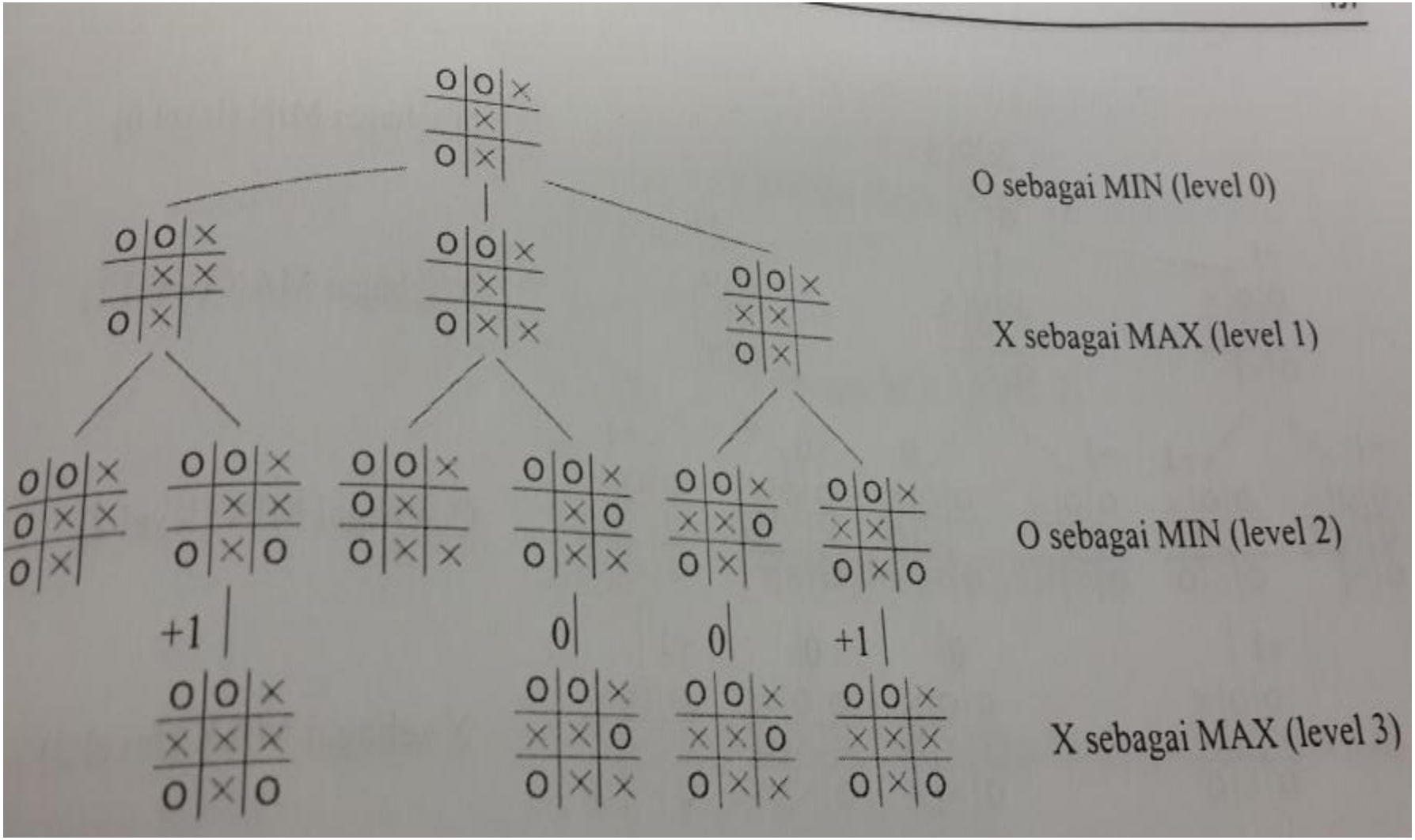
1. Misalkan ada 2 pemain yang terlibat, kita namakan MAX dan MIN.
2. Lalu sebuah pohon pencarian dibangkitkan secara *depth-first-search* dari posisi awal permainan hingga akhir permainan.
3. Dari sudut pandang MAX, akan dicari posisi terakhir yang paling menguntungkan bagi MAX. MAX akan mengambil nilai maksimum dari pohon pencarian yang dibangkitkan pada simpul terakhir. Sebaliknya, MIN akan menangkis serangan MAX dengan mengambil nilai minimum pada posisi akhir permainan, yang akan meminimalisasi serangan MAX.

Contoh ada pada tic tac toe



X : komputer O : Human

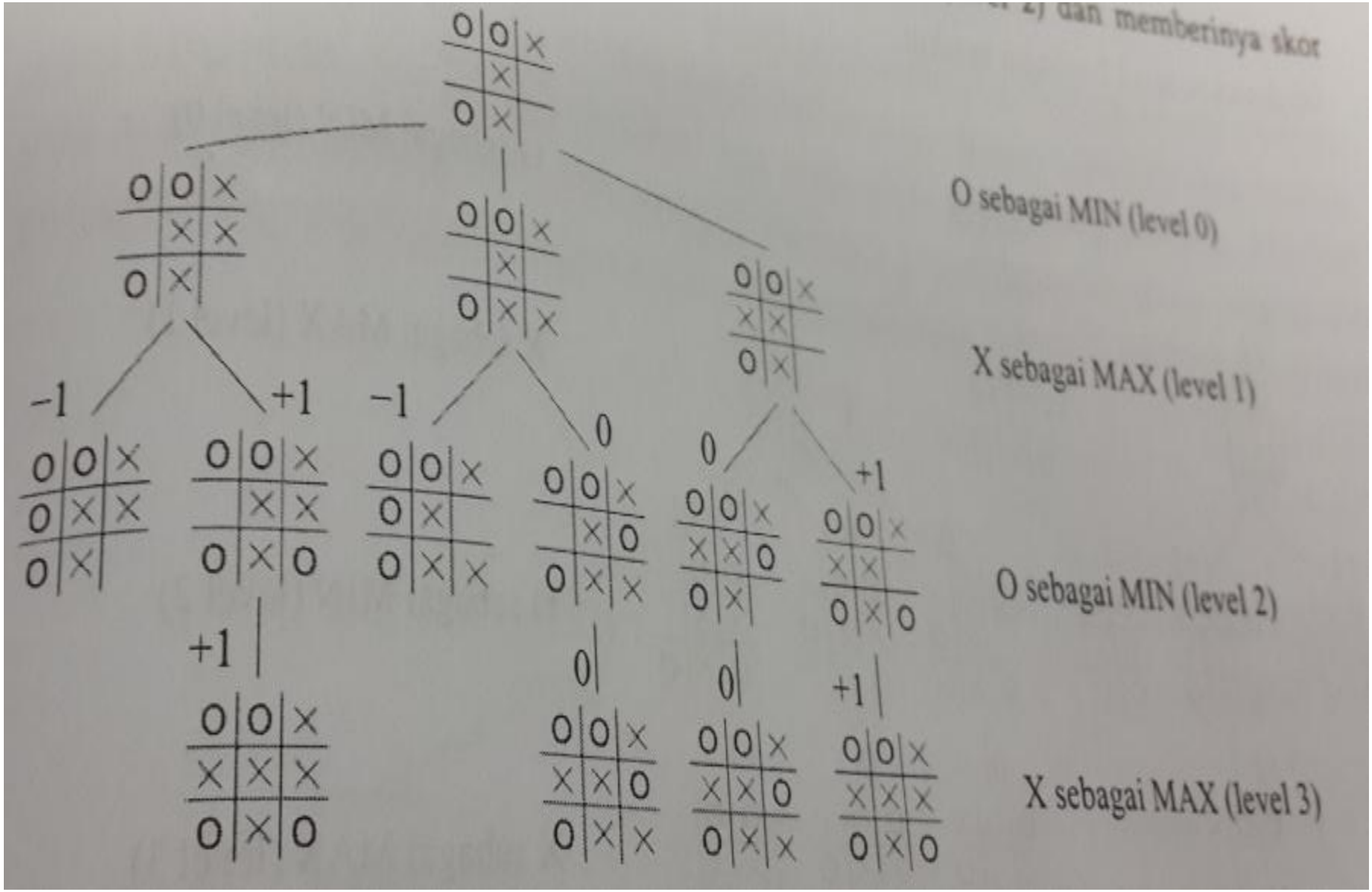
Root : posisi setelah human bergerak

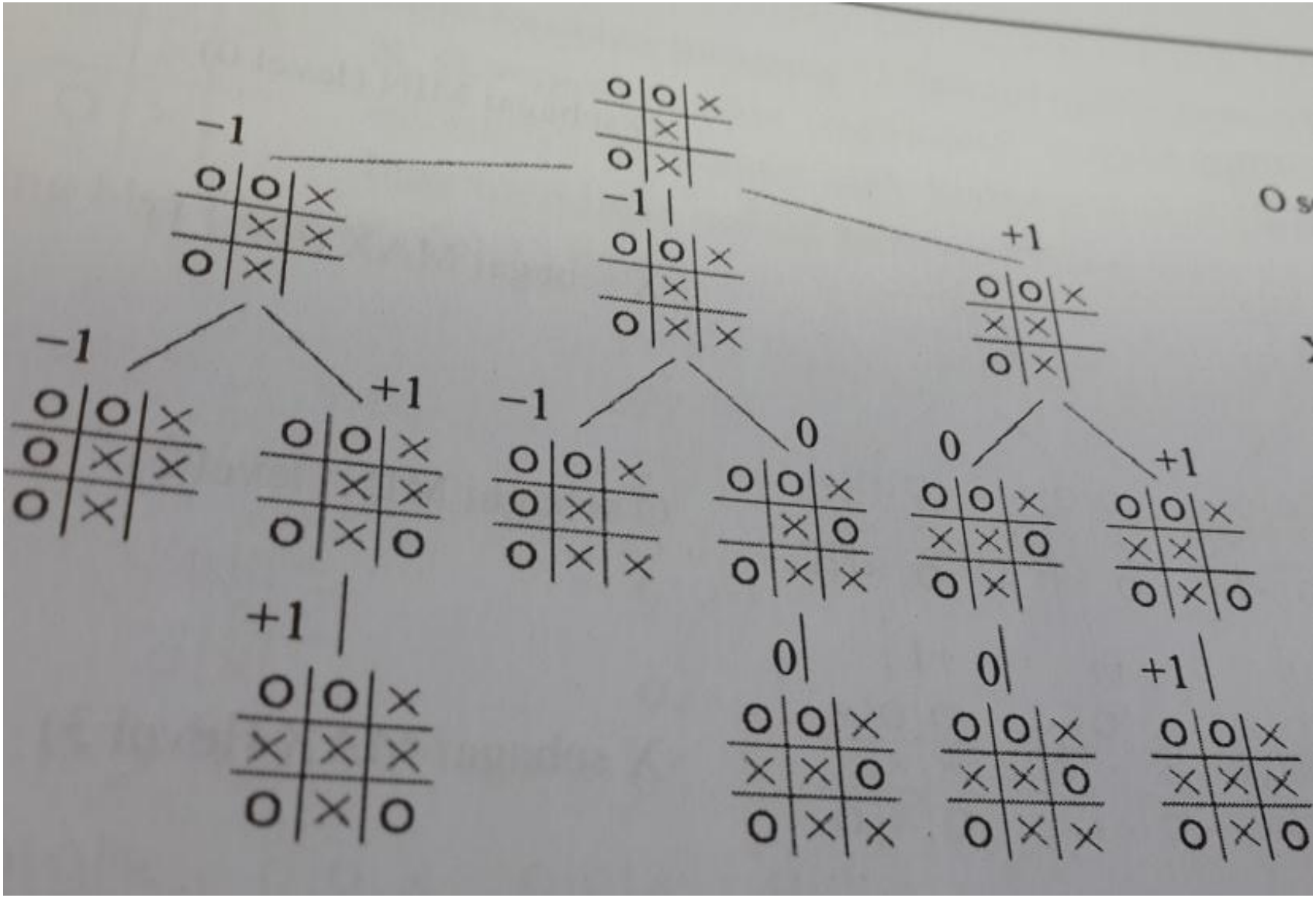


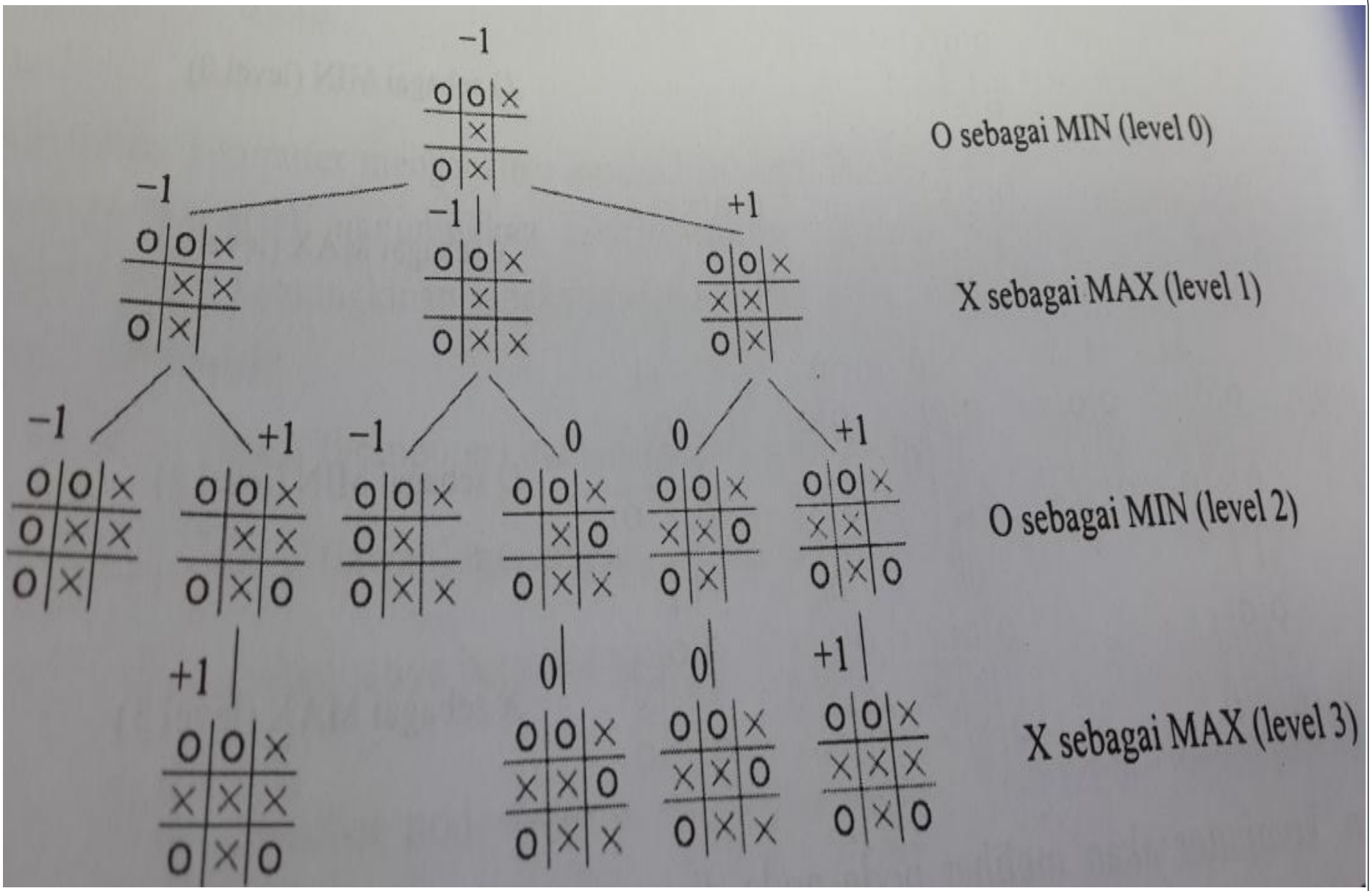
+1 = Komputer Menang

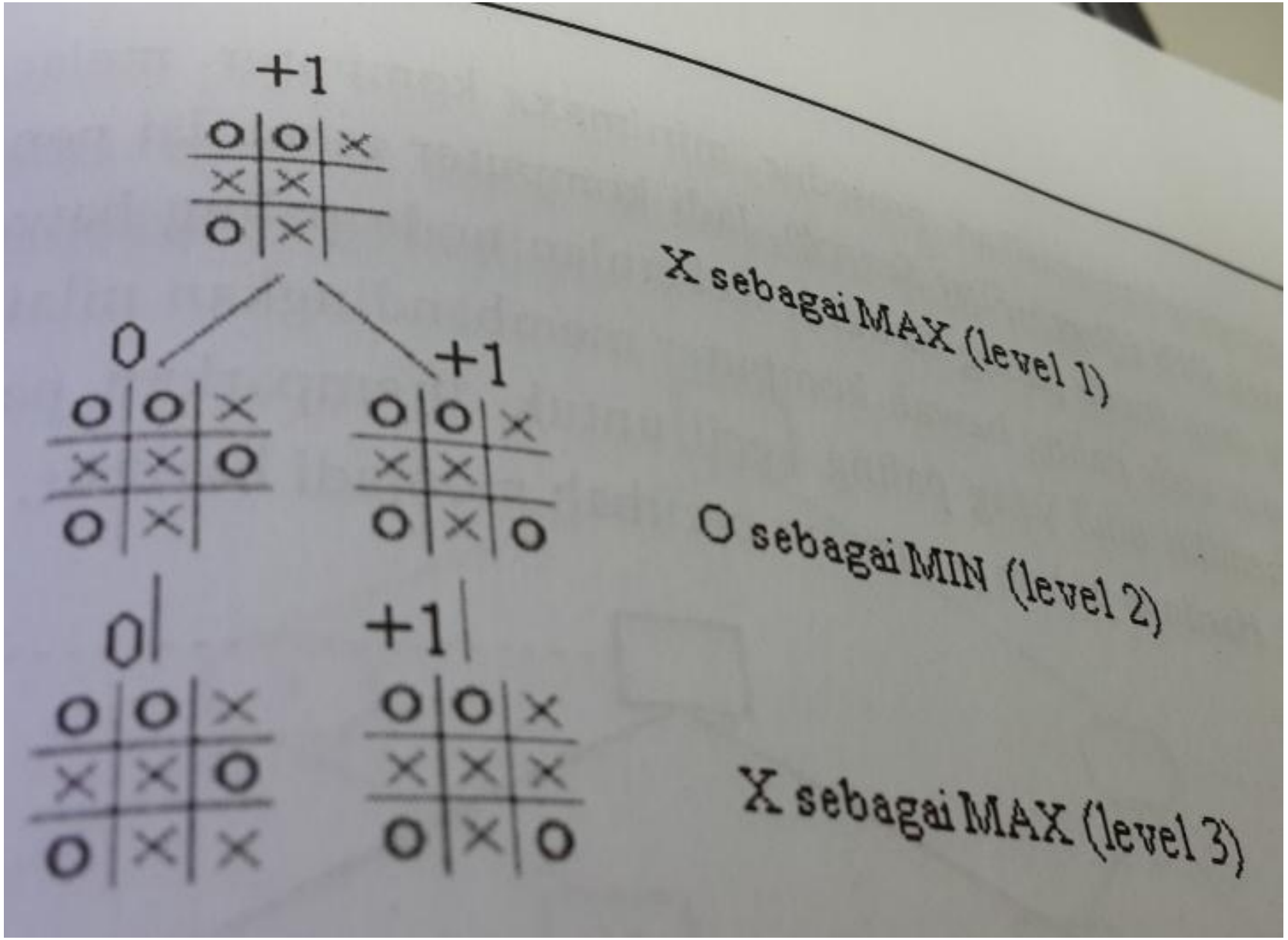
-1 = Human Menang

0 = Seri


O sebagai MIN (level 0)
X sebagai MAX (level 1)
O sebagai MIN (level 2)
X sebagai MAX (level 3)
-1
+1
-1
0
0
+1
+1
0
0
+1


−1
−1
+1
+1
−1
0
+1
0
0
+1


O sebagai MIN (level 0)
X sebagai MAX (level 1)
O sebagai MIN (level 2)
X sebagai MAX (level 3)


+1
0
+1
0
+1
X sebagai MAX (level 1)
O sebagai MIN (level 2)
X sebagai MAX (level 3)

Karekteristik Minimax

- Completeness : dikatakan lengkap jika pohon pencaraiannya berhingga (terbatas).
- Optimalitas : Algoritma minimax akan optimal jika melawan pemain yang juga memiliki langkah optimal
- Kompleksitas waktu : kompleksitas waktunya besar, karena semua node tree di eksplor $O(b^m)$, contoh pada permainan catur b adalah faktor percabangan (node yang terbentuk), m adalah kedalaman dari pohon pencarian.
- Kompleksitas ruang : O(bm)

Minimax

- Kelemahan algoritma Minimax yaitu tidak mampu memroses data dengan ukuran masukan yang besar, karena proses untuk membangun pohon pencarian dengan algoritma ini memiliki kompleksitas algoritma eksponensial. Maka dari itu, diperlukan optimasi dalam algoritma ini agar tidak semua simpul dibangkitkan.

Alfa-Beta Pruning

- Optimalisasi algoritma minimax
- Memangkas node yang tidak perlu di telusuri
- Nilai alfa adalah nilai sementara yang diasosiasikan dengan node Max. nilai awalnya adalah $\alpha = -\infty$
- Nilai Beta adalah nilai sementara yang diasosiasikan dengan node Min . nilai awalnya adalah $\beta = \infty$

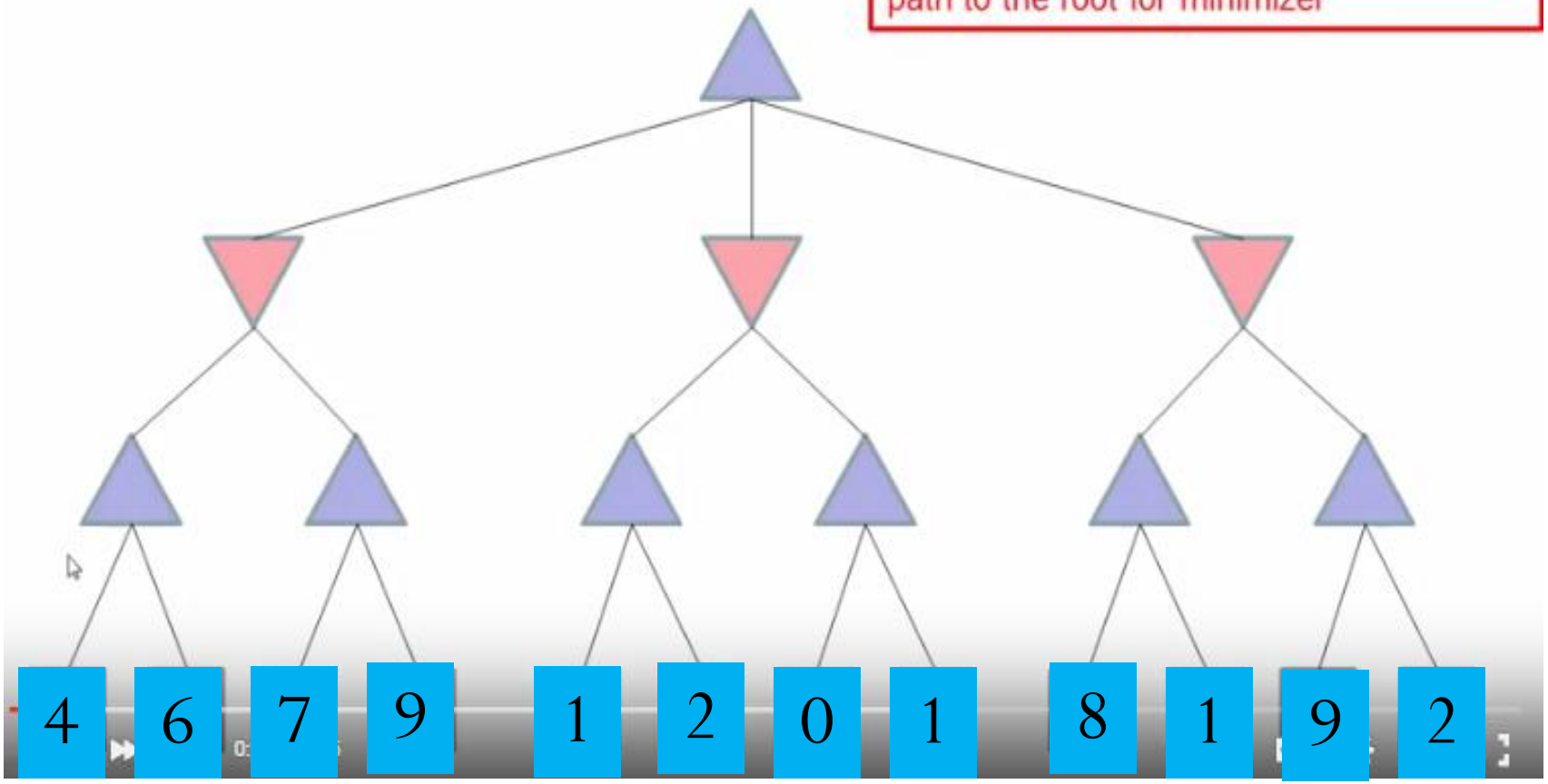
Alfa – Beta Pruning

= Maximum

= Minimum

Alpha-Beta Example

Alpha = best already explored option along path to the root for maximizer
Beta = best already explored option along path to the root for minimizer



Alpha-Beta Example

Node yang belum dikunjungi :

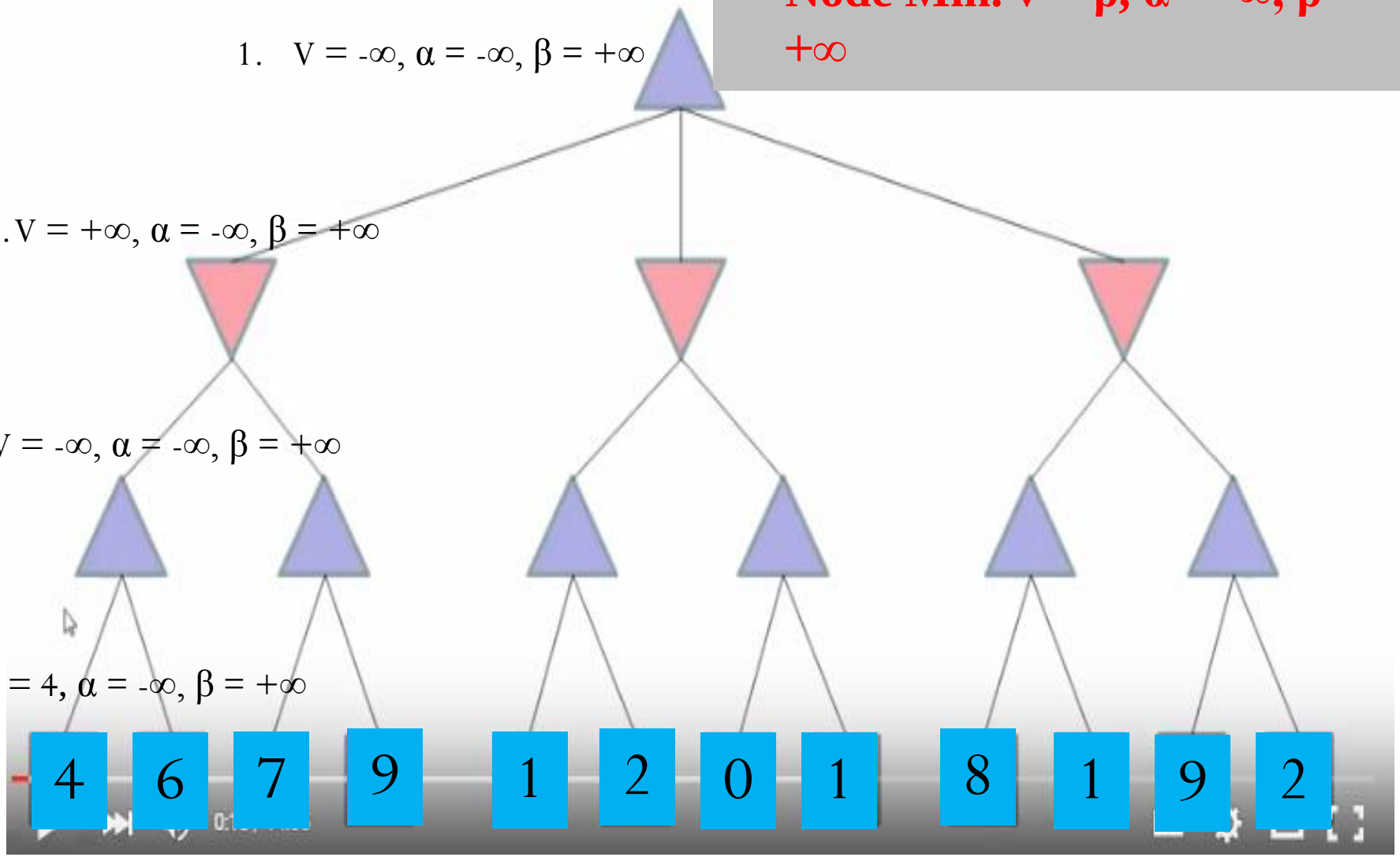
- **Node Max. v = α, α = -∞, β = +∞**
- **Node Min. v = β, α = -∞, β = +∞**

1. V = -∞, α = -∞, β = +∞

2. V = +∞, α = -∞, β = +∞

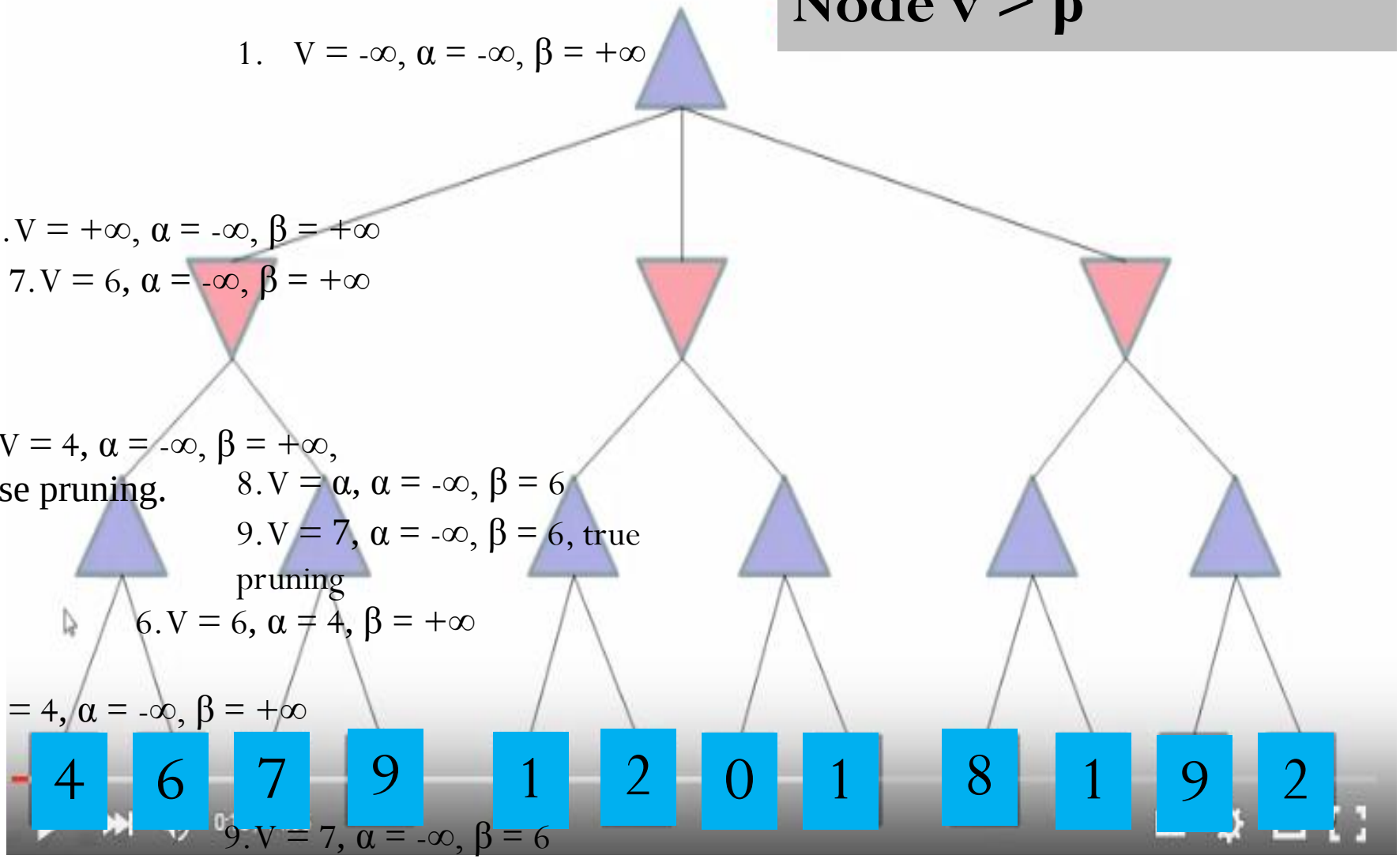
3. V = -∞, α = -∞, β = +∞

4. V = 4, α = -∞, β = +∞



Alpha-Beta Example

Pruning dilakukan jika
Node v > β



Alfa – Beta Pruning