Theory and Practice of Artificial Intelligence

Games

Daniel Polani

School of Computer Science
University of Hertfordshire

March 9, 2017

Games

More Precisely:

- two-person (not multi-person; no gang-ups)
- perfect information (no card games)
- deterministic (no backgammon)
- alternating moves (no rock/scissors/paper)
- zero-sum (no prisoner's dilemma)

games

Game Structure

Conditions: game is over when *terminal* position reached where game ends (no successor moves).

Possible Outcomes: consider *win/loss/draw*. Other, intermediate outcomes also possible.

Game State Structure

Game:

- game position
- terminal won position
- terminal lost
- non-terminal won
- *us-to-move* (player A)
- *them-to-move* (player B)

Position Utilities

Motivation: since, in general, game trees are too big to be completely solved, use a *utility* (value) function to indicate which positions are more promising than another.

Implication: quality of a game state characterized by its value (utility) U, a real-valued number

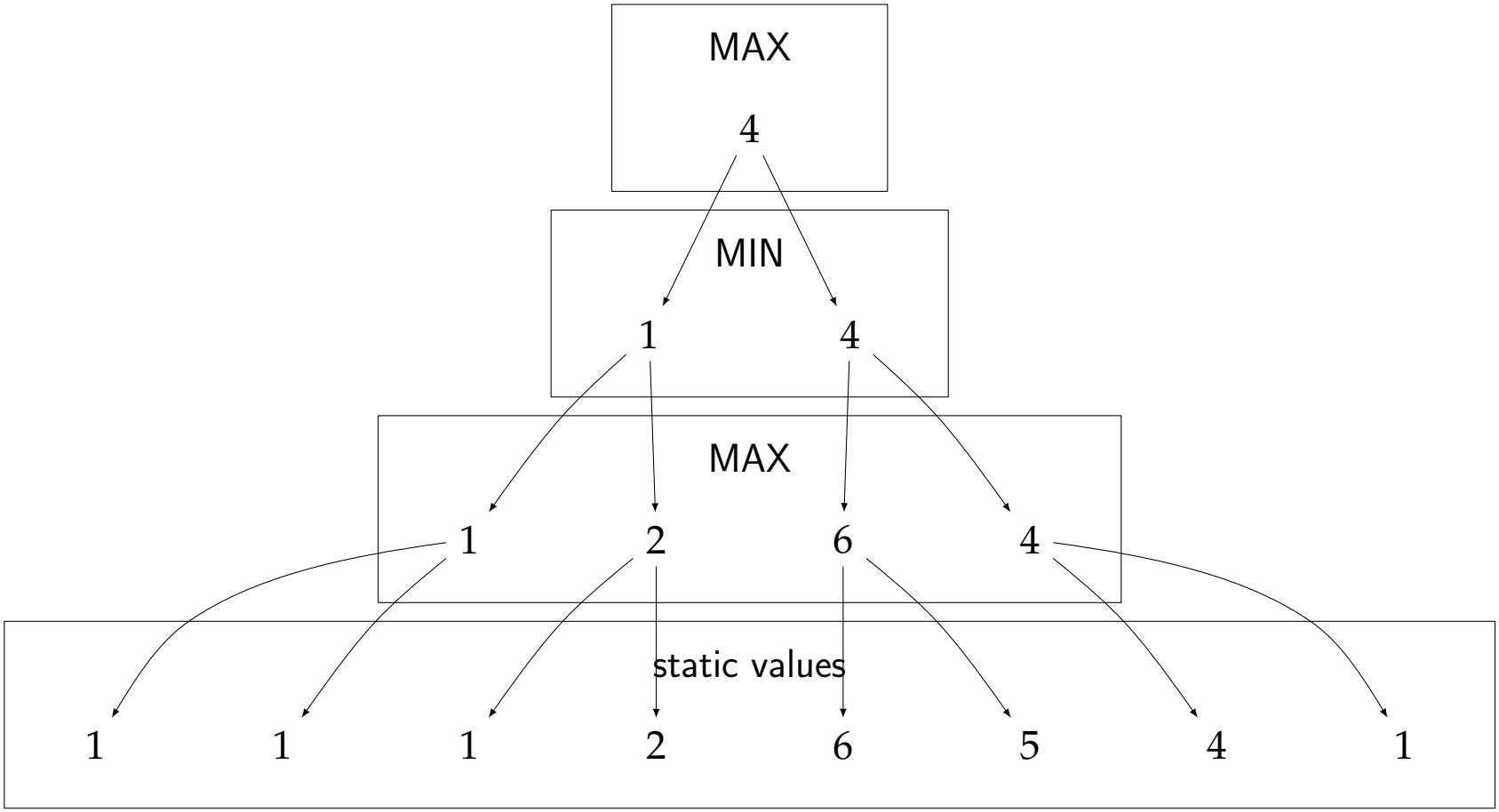
Note: "promising" subtrees are indicated by a high value of U for starting states.

Position Utilities II

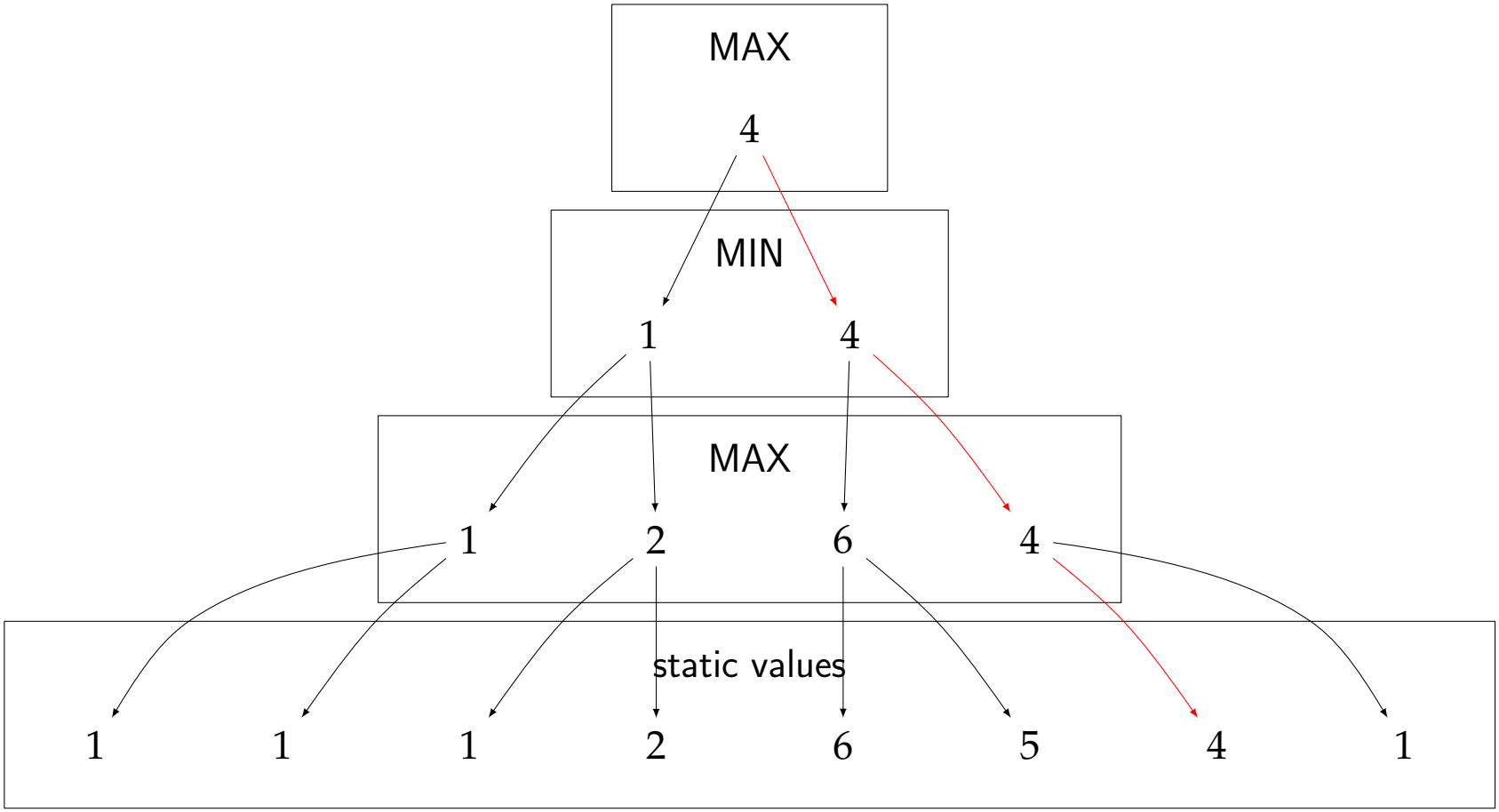
Note: the *true* value U of a position indicates the state of the position won/lost/draw, e.g.

- $U = 100$: current position allows player A to win (on optimal game from both sides)
- $U = -100$: current position is lost for player A (on optimal game from both sides)
- $U = 0$: position is a draw (no player can force a win)

Minimax Principle



Minimax Principle (Main Variation)



Minimax view of utilities

Consider: $U(P)$, the utility of a position

Let: $S(P) = \{P_1, P_2, \ldots, P_n\}$ be the set of successors for position P

Minimax Utility: define

$$U(P) = \begin{cases} U_{\text{static}}(P) & \text{if } P \text{ terminal, i.e. } S(P) = \{\} \\ \max\limits_{P_i \in S(P)} U(P_i) & \text{if } P \text{ is a MAX-to-move position} \\ \min\limits_{P_i \in S(P)} U(P_i) & \text{if } P \text{ is a MIN-to-move position} \end{cases}$$

The Alpha-Beta Algorithm

Observation:

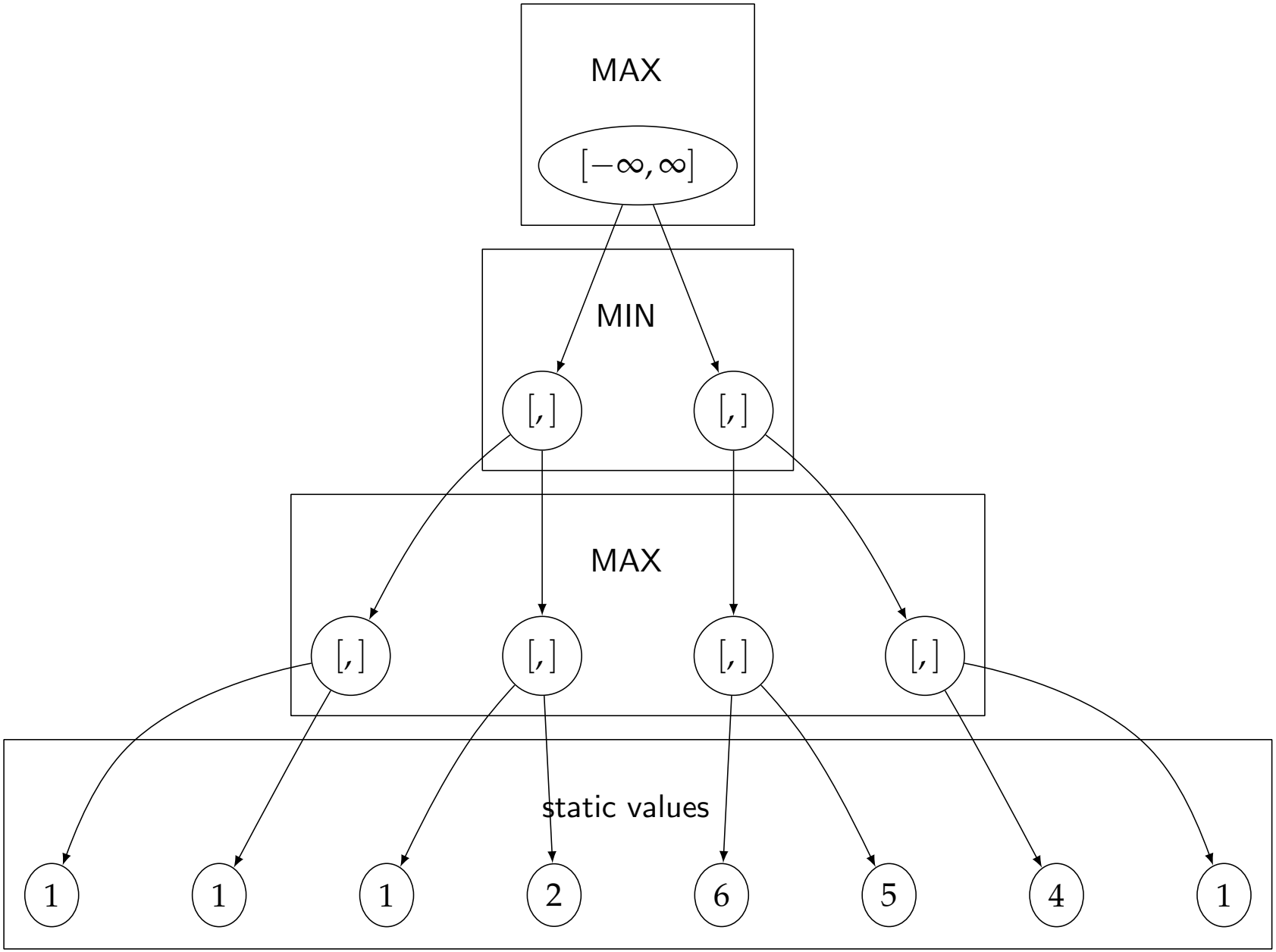
- sometimes we know a move is not good and will never be covered
- in that case, the exact utility of the node is not needed

α-β principle:

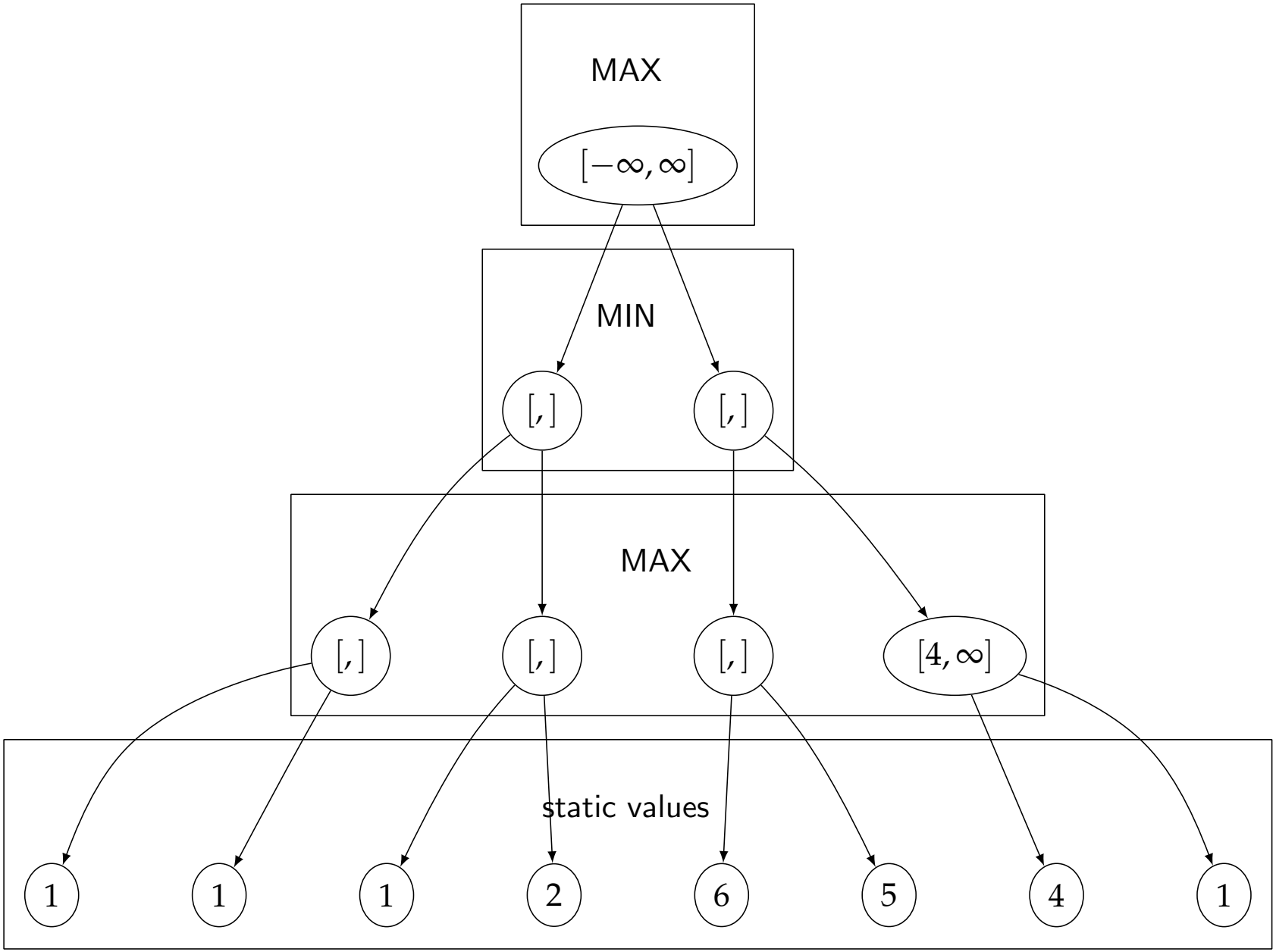
- search for the utility of a position but only if in the interval $[\alpha, \beta]$
- if it is outside, its exact value is not important, we will be prevented from taking that path anyway

Illustration: see following slides

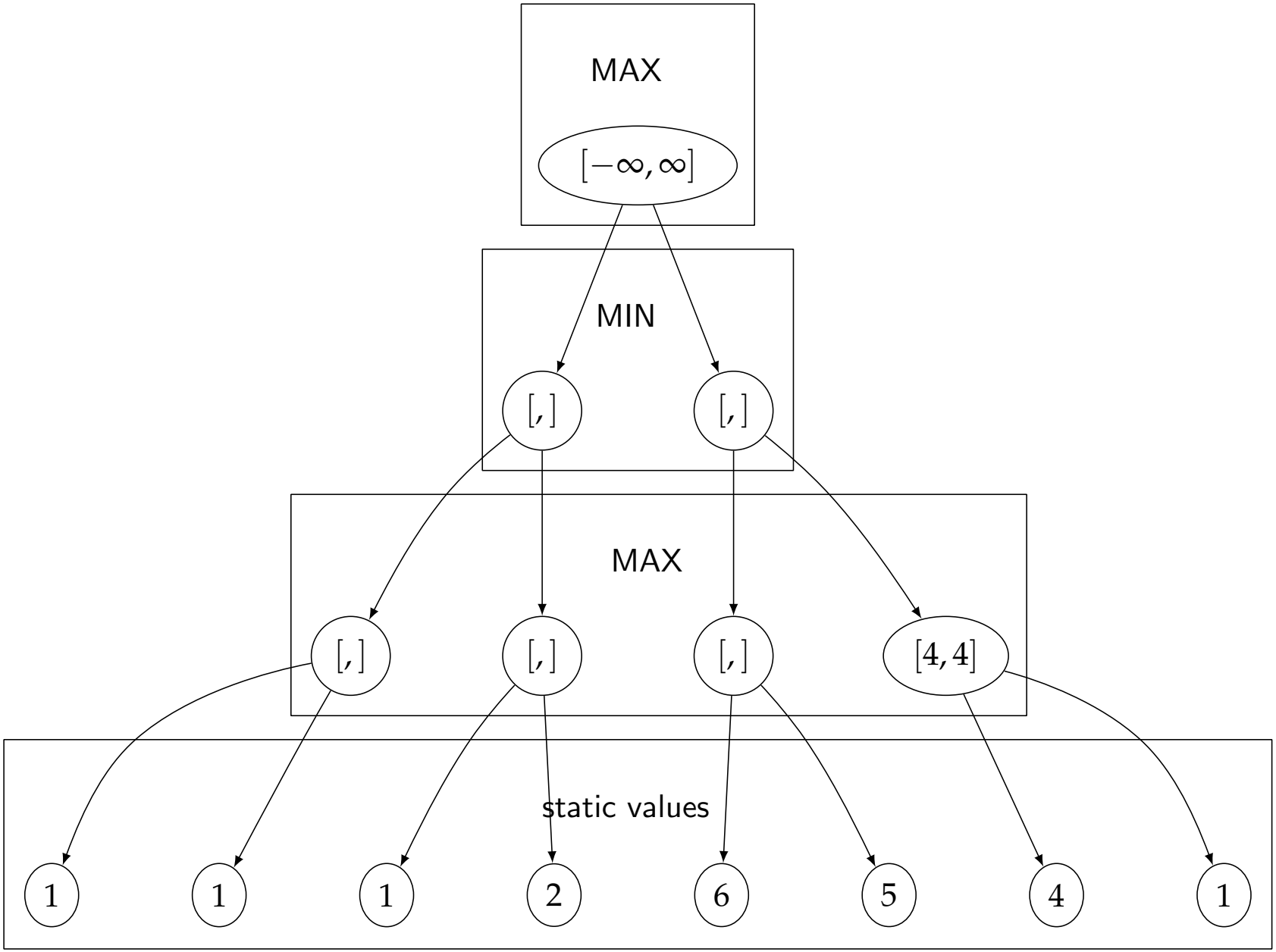
The Alpha-Beta Algorithm



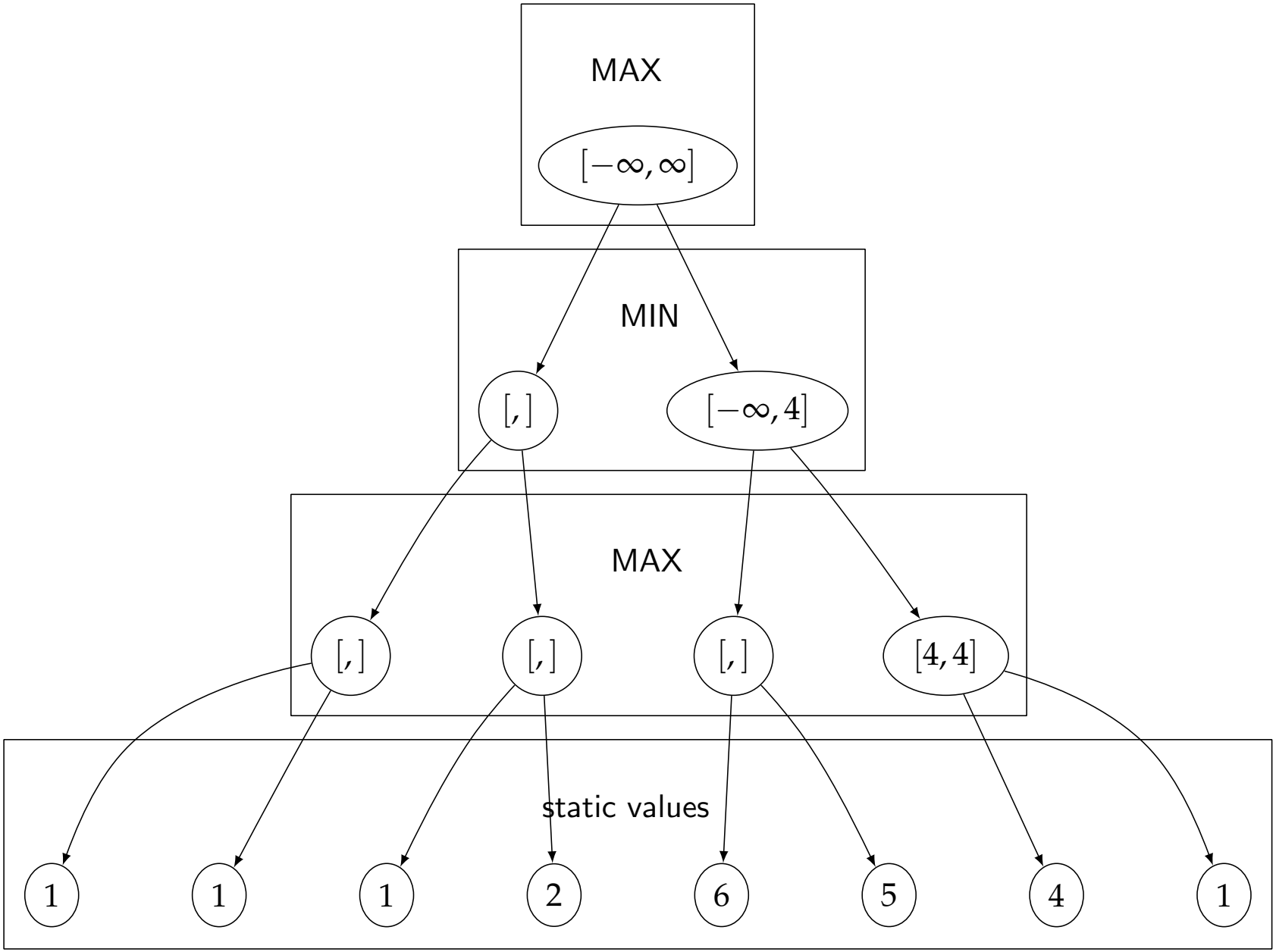
The Alpha-Beta Algorithm



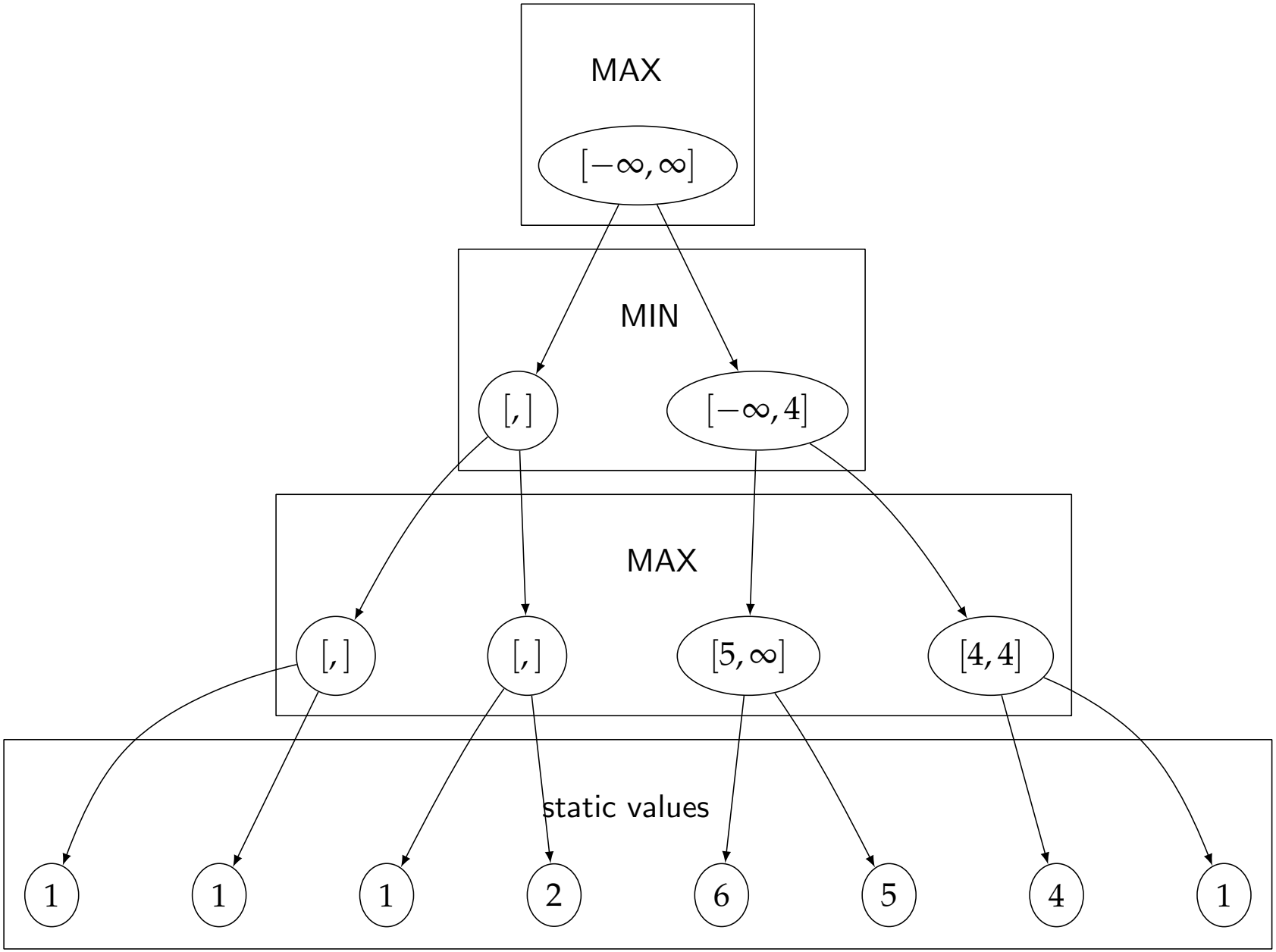
The Alpha-Beta Algorithm



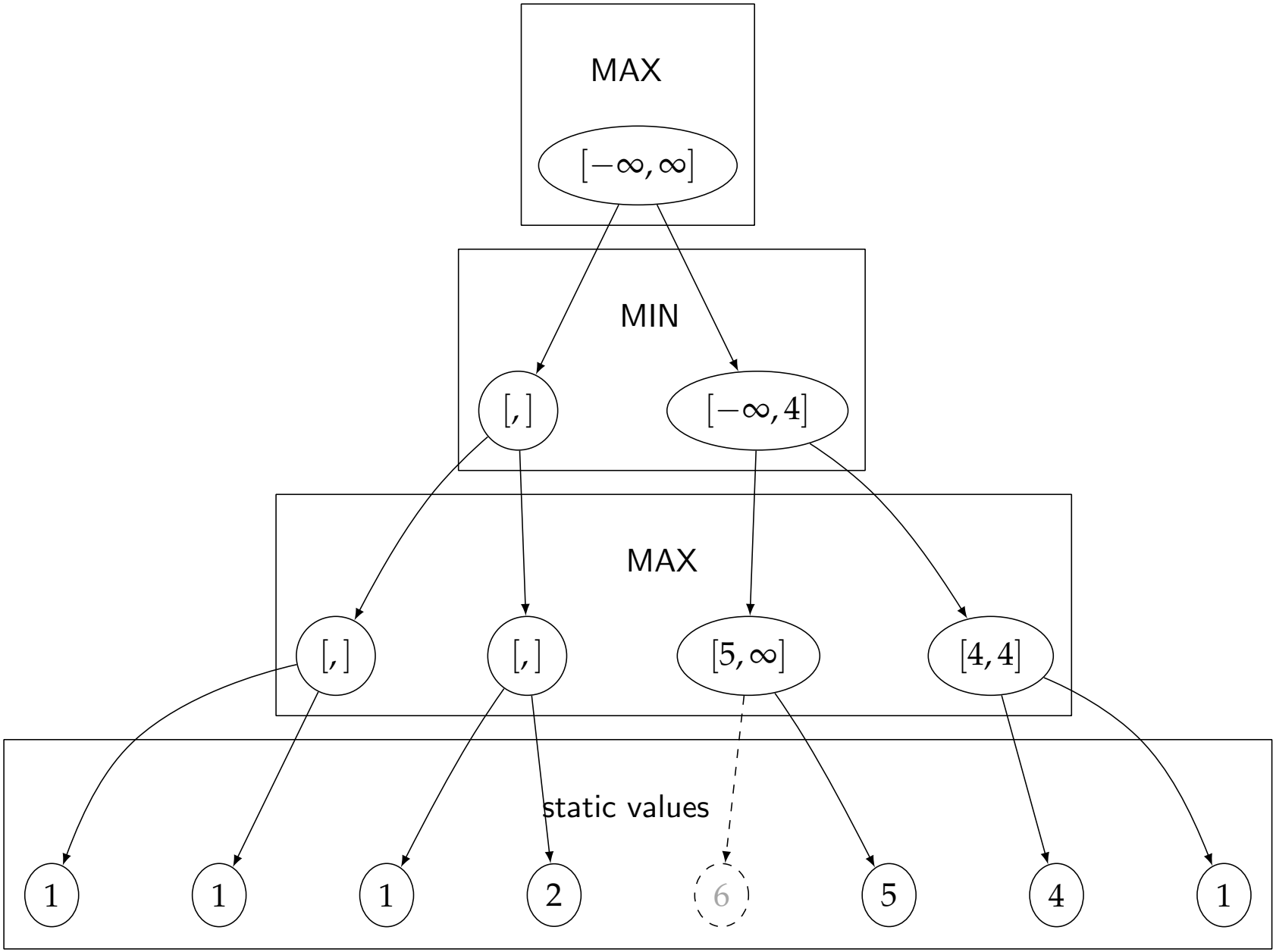
The Alpha-Beta Algorithm



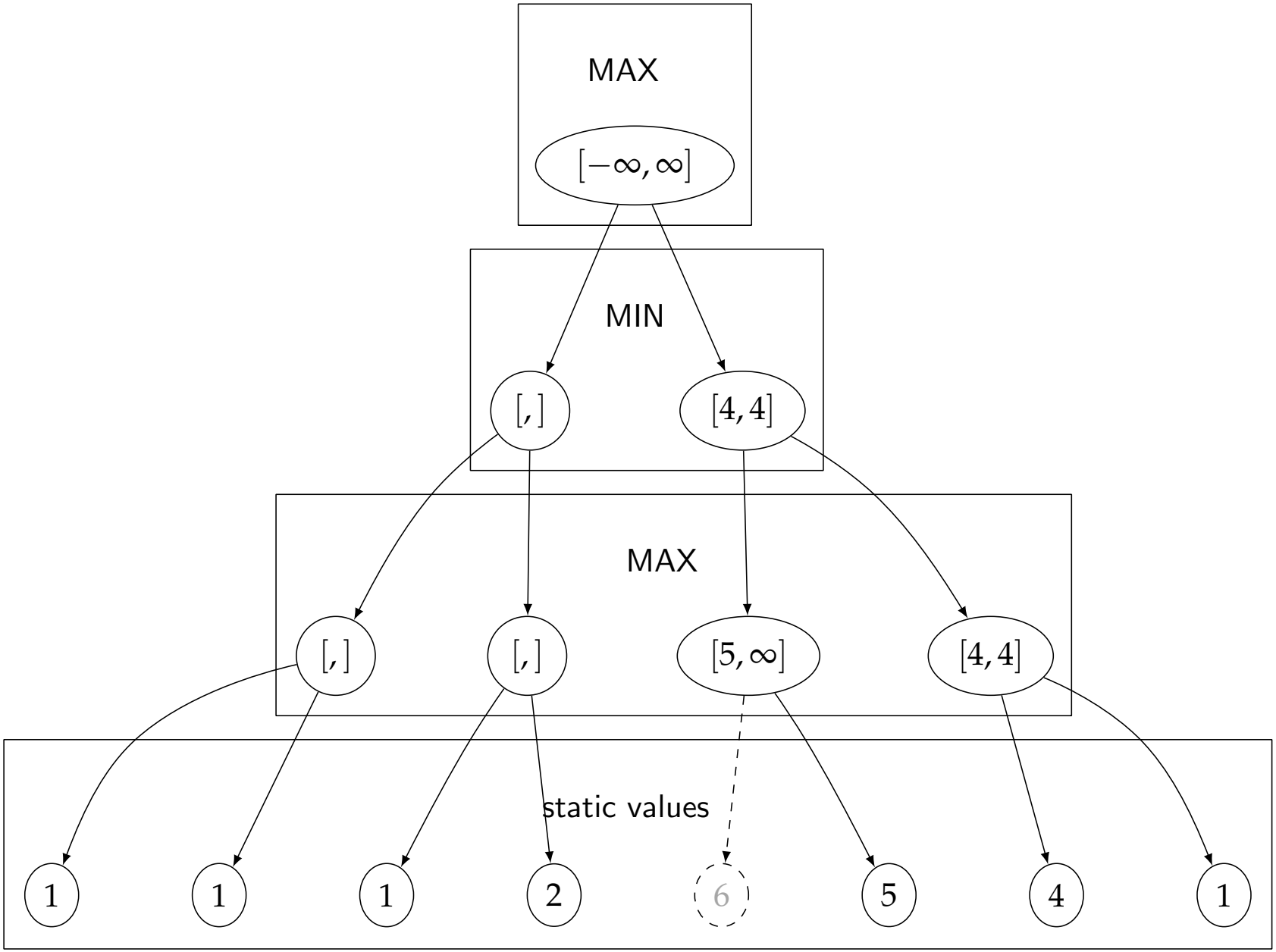
The Alpha-Beta Algorithm



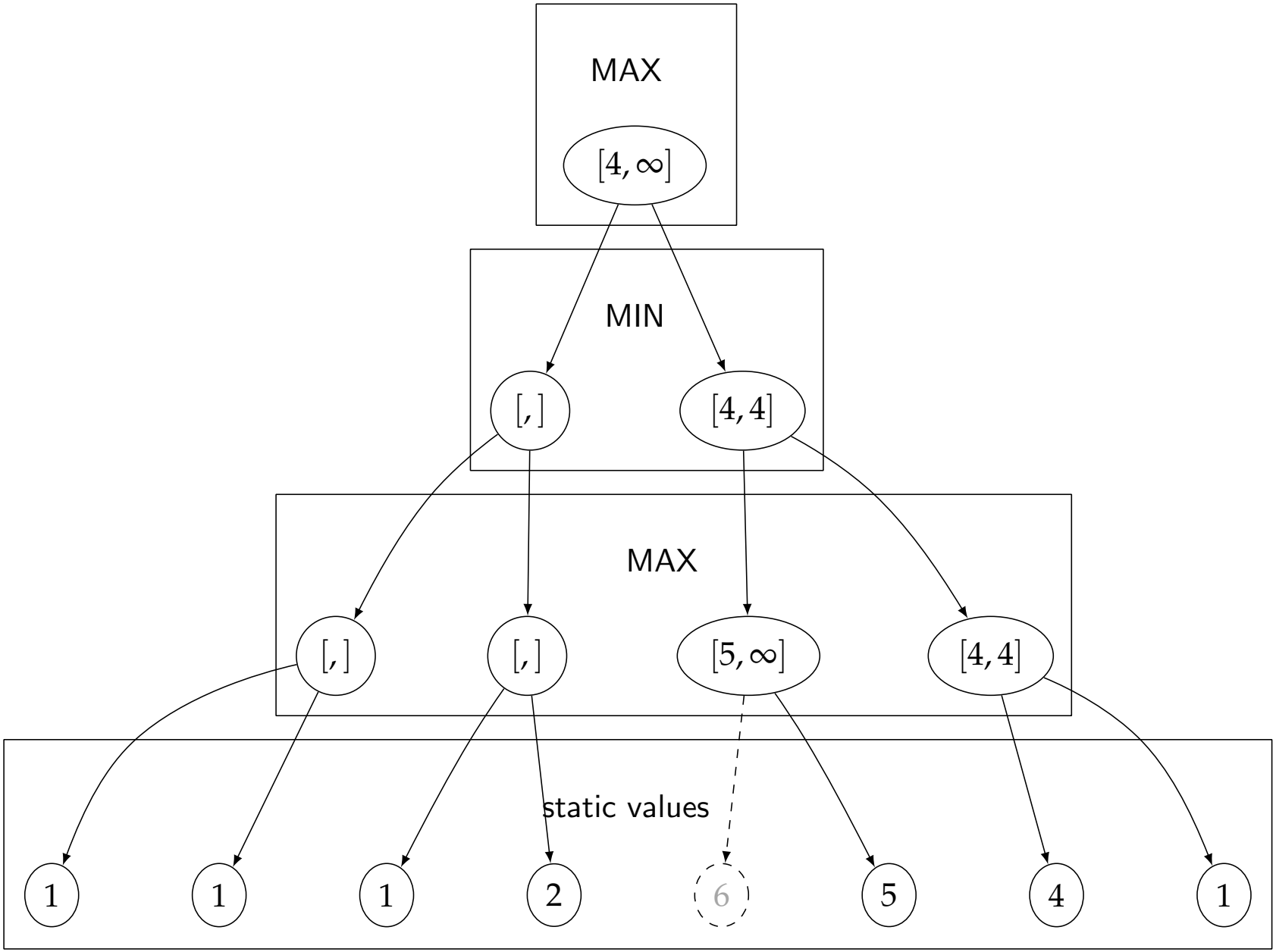
The Alpha-Beta Algorithm



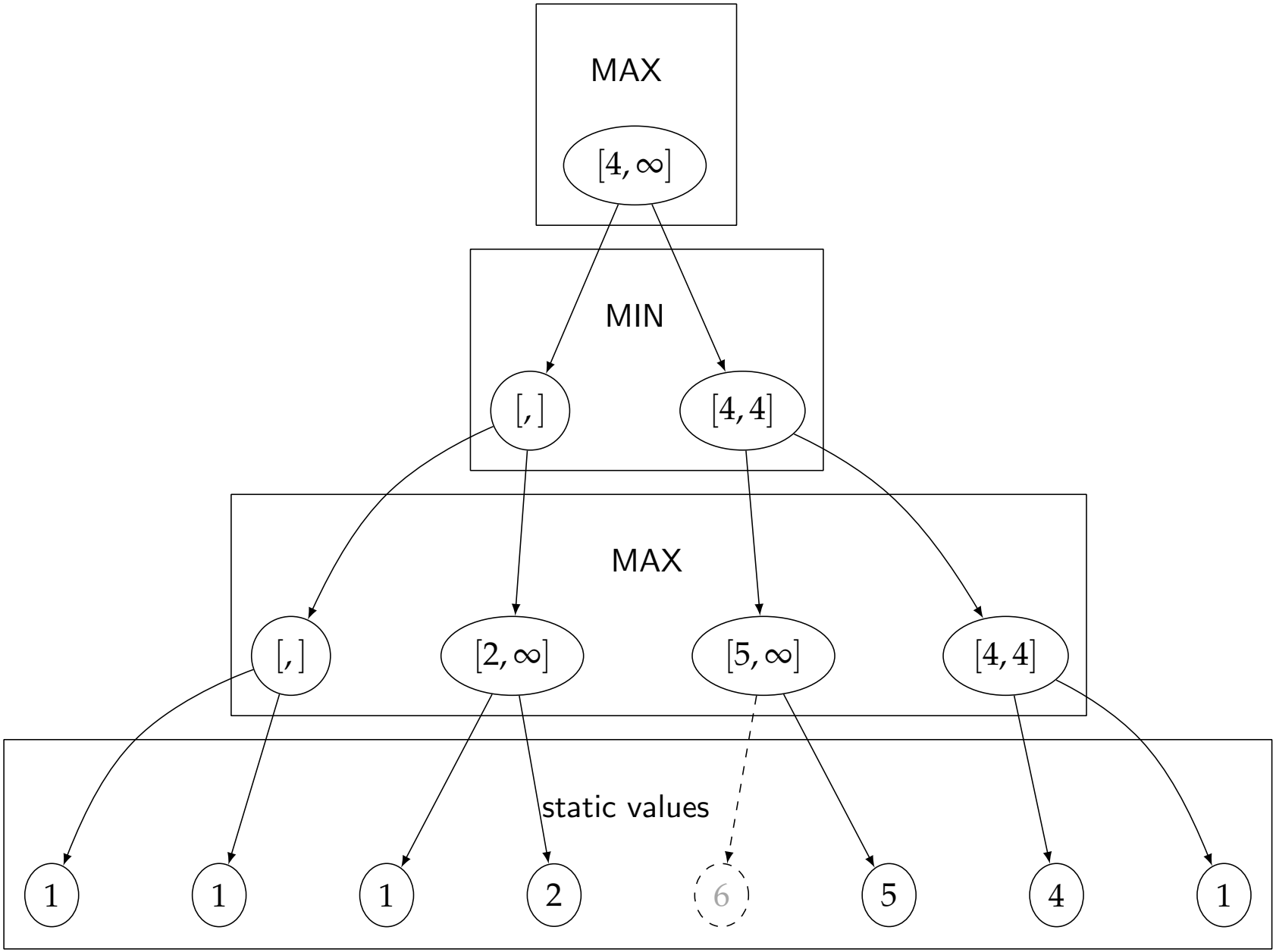
The Alpha-Beta Algorithm



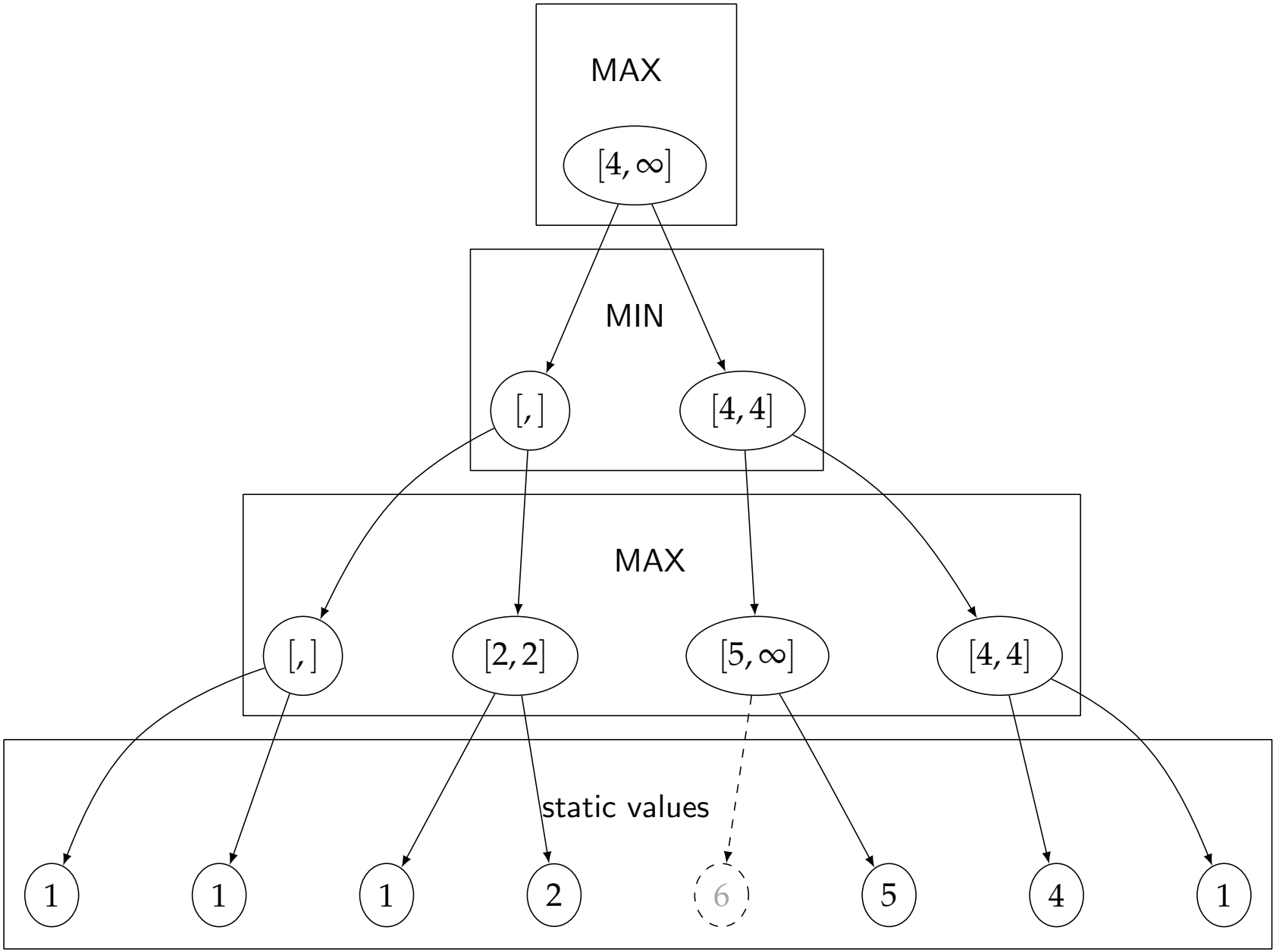
The Alpha-Beta Algorithm



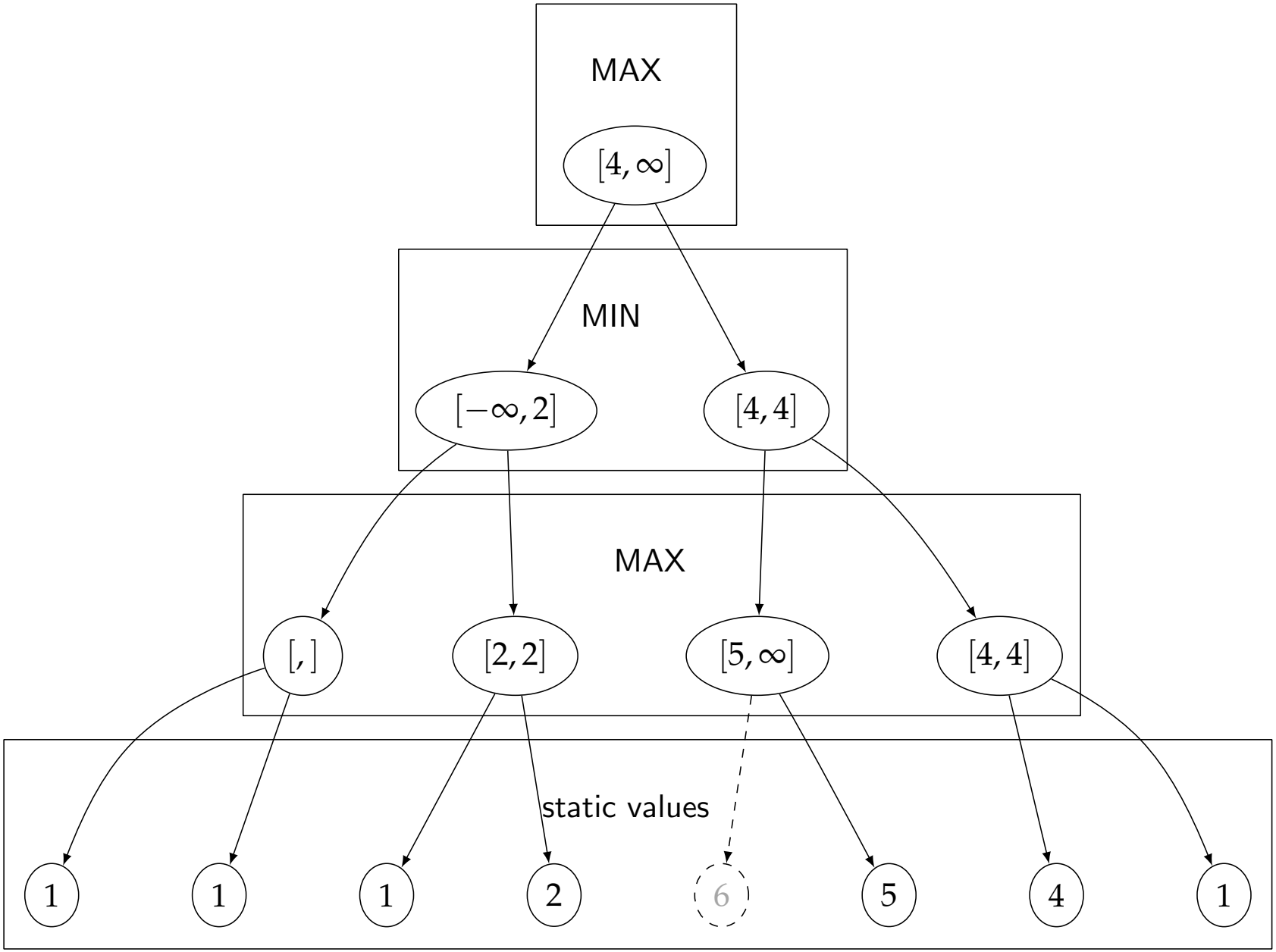
The Alpha-Beta Algorithm



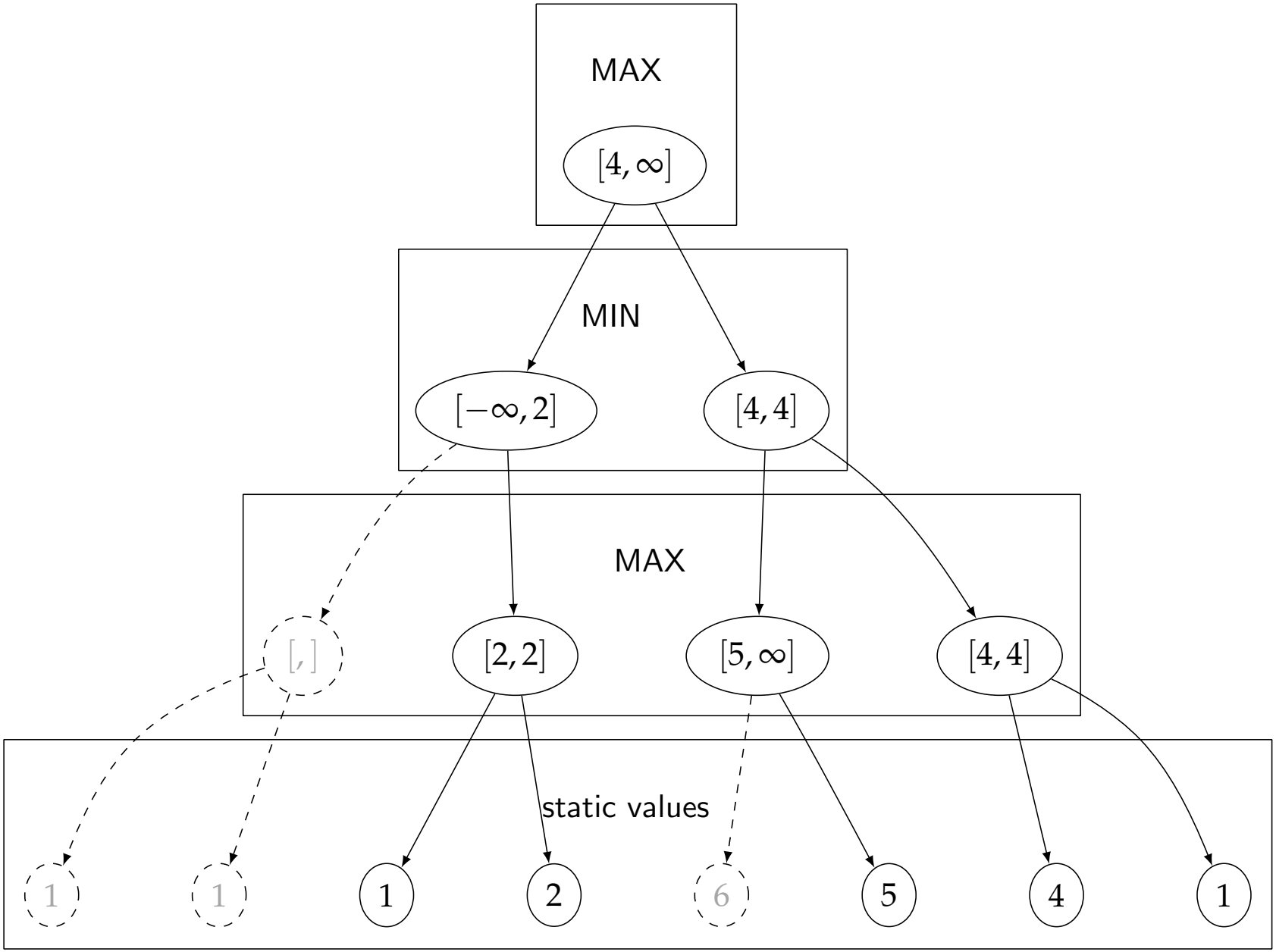
The Alpha-Beta Algorithm



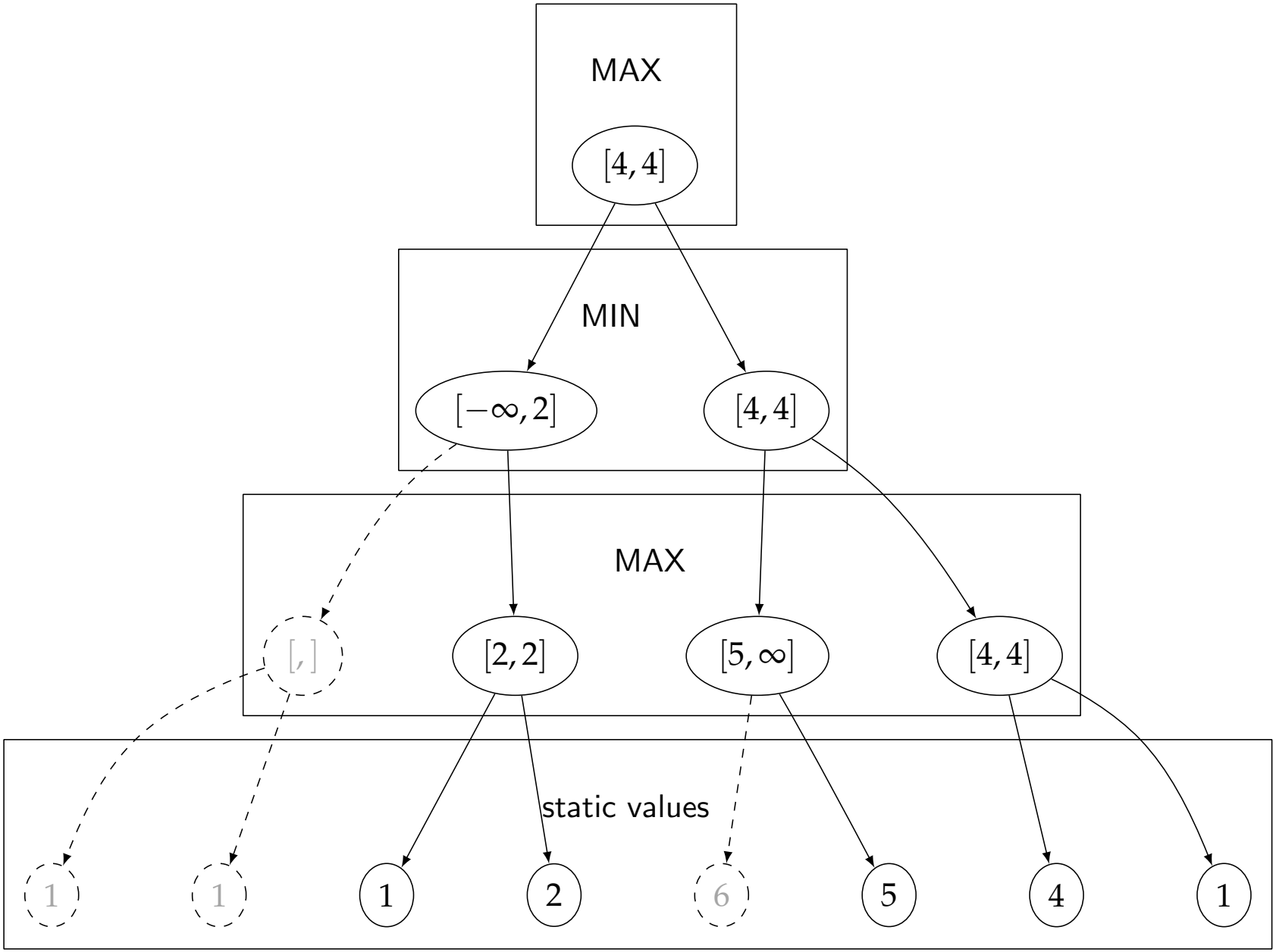
The Alpha-Beta Algorithm



The Alpha-Beta Algorithm



The Alpha-Beta Algorithm



Alpha-Beta Algorithm: Properties

α: worst guaranteed utility for MAX
(and best achievable value for MIN)

β: worst guaranteed utility for MIN
(and best achievable value for MAX)

Good Enough Utility: a utility $U(P, \alpha, \beta)$ is a utility such that

$$\begin{aligned} U(P,\alpha,\beta) &< \alpha && \text{if } U(P) < \alpha \\ U(P,\alpha,\beta) &= U(P) && \text{if } \alpha \leq U(P) \leq \beta \\ U(P,\alpha,\beta) &> \beta && \text{if } U(P) > \beta\ . \end{aligned}$$

In Particular: $U(P, -\infty, \infty) = U(P)$

Remark: in the best case, this reduces the search branching factor from b for minimax to $\sqrt{b}$

Thus: can search twice as deeply as with minimax with the same evaluation effort

Further Improvements

1. limitation of move selection
2. heuristic value function (cutoff before final state)
3. quiescence heuristics

Further Improvements

1. limitation of move selection
2. heuristic value function (cutoff before final state)
3. quiescence heuristics
4. **endgame algorithm**

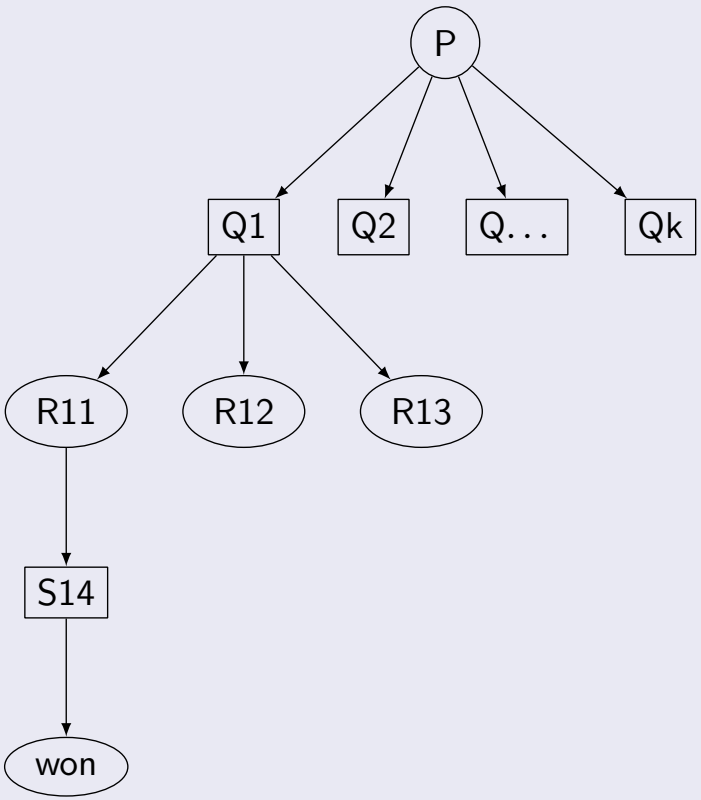
Further Improvements

1. limitation of move selection
2. heuristic value function (cutoff before final state)
3. quiescence heuristics
4. **endgame algorithm**
5. **UCT Monte Carlo Tree Search**

Game-Playing to the End: Idea

End Games: consider

- game with only *win/loss*
- 2 players **us** and **them**
- playing alternatively
- solution: win for **us**



Interpretation: game is won if solution tree exists, i.e. tree begins with an

us node: there is a choice for **us** leading to an

them node: such that all possible choices for **them** lead to an

us node: and so on until

Goal: successful solution (**win**) is found

Interpretation

It means: **us** has won (solution tree) if it is either

- in a **winning position** or it can always choose a move leading
- to a **losing** position of **them**; i.e. a position such that all moves that **them** can choose lead
- to a winning position of **us** (i.e. again to a solution tree).

Note: **us** does not have to have a solution tree. Either

- **them** could have a solution tree (in which **us** loses)
- or neither of them have, so none of the players can force a win.

Yes, I treat **us** as singular player and not as *pluralis majestatis*.

Endgame Algorithm

Endgame Algorithm: for **us**

1. consider final (0-step) winning positions for **us**
2. compute 1-step losing positions for **them**, i.e.
   - all positions for **them** from which
   - **all** immediate successors lead
   - to a 0-step winning position for **us**
3. compute 2-step winning positions for **us**, i.e.
   - all positions where **us** can choose
   - *one* immediate successor to lead
   - to a 1-step losing position for **them**
4. compute 3-step losing positions for **them**, i.e.
   - all positions for **them** where
   - *all* successors lead
   - to a less-than-3 (i.e. 2- or 0-) winning position for **us**.
5. and so on, until no more new positions are collected or maximum depth are exhausted

Result: if no maximum depth limit, the final outcome is a

- list of winning positions for **us** (with maximum depths)
- a list of losing positions for **them** (with maximum depths)
- and a list of tied positions