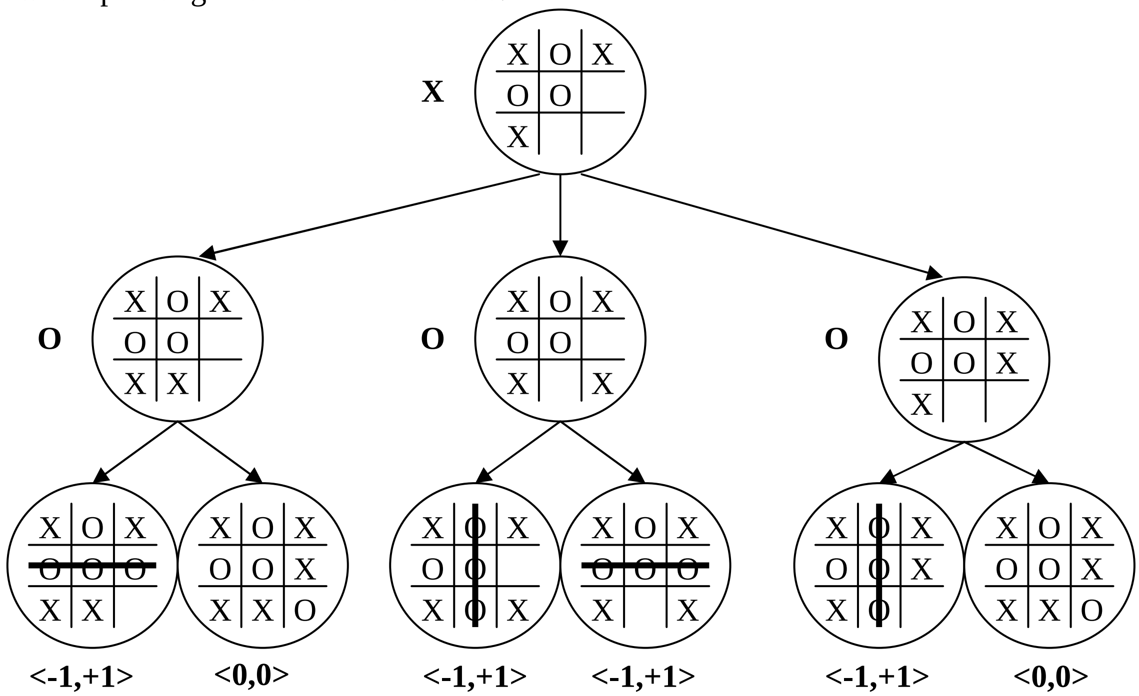
adversarial search (games) – competitive multiagent environments (agent's have conflicting goals). In particular, adversarial search is mixture of *search* and *game theory*. The typical game is a *deterministic*, *turn-taking*, *two-player* **zero-sum** game of **perfect information**. These games are a sequence of decisions that reach a *terminal state*. Below is a partial game tree for tic-tac-toe:



- **game tree** – a representation that represents all legal sequences of decisions.
  - **root –** the *initial state* of the game (with a starting player).
  - **(internal) nodes –** represents decision made by one of the players. The node is labeled by the player making the decision (*Max/Min*).
  - **edges –** legal choices for a given decision in the tree. These are specified by a *successor function* that lists legal (*move*, *state*) pairs.
  - **terminal node –** an ending of the game giving a *utility* to each player.
    - **utility function –** maps a terminal state to a value.
- **optimal strategy** – a contingent strategy that leads to an outcome at least as good as any other strategy by assuming the opponent is infallible.
  - minimax algorithm – finds an optimal strategy by depth-first exhaustive search which annotates each node of the tree with a **minimax-value**:

$$\text{minimax-value}(n) = \begin{cases} \text{utility}(n) & n \in \mathit{Terminal} \\ \max\limits_{s \in child(n)} \text{minimax-value}(s) & n \in \mathit{MAX} \\ \min\limits_{s \in child(n)} \text{minimax-value}(s) & n \in \mathit{MIN} \end{cases}$$

- **alpha-beta pruning** – a modified minimax search that prunes branches that cannot influence the final result.
  - α – the maximum value so far at any choice point along the path for MAX
  - β - the minimum value so far at any choice point along the path for MIN

<u>Stopping search prematurely</u> – time limits prevent full exploration of the game tree.

- **evaluation function** – a heuristic for accessing the utility of a nonterminal game state; that is, it returns an estimate of the expected value of a state.
  - **features** – elements of the state that indicate its strength.
    - features form *categories* (*equivalence classes*) among states.
    - many evaluation functions combine numerical contributions from each feature as an estimate (e.g. weighted linear function).
- **cutting-off search** – determine a reasonable time to stop search (e.g. *iterative deepening* explores deeper until time elapses).
  - evaluation function should only be applied to positions that are unlikely to have major changes in the near future (*quiescent*).
  - **horizon effect** – an unavoidable damaging move looms on the horizon.

<u>Games of Chance</u>

- **chance nodes** – nodes (denoted by circles) indicating an element of chance is introduced and arcs from this node are probabilistic transitions
  - The minimax algorithm is identical & chance nodes are *expected values:*

    $$\text{expectiminimax}(n) = \sum_{s \in child(n)} P(s) \cdot \text{expectiminimax}(s) \qquad n \in Chance$$

    - In games of chance, the evaluation function *must be a* ***positive linear transform*** *of the probability of winning from a position.*
  - Pruning of chance nodes is possible if bounds can be placed on possible values (thereby bounding the possible values of the average).

<u>Games of Chance with *imperfect information*</u>

- **averaging over clairvoyancy –** the strategy of computing optimal moves by averaging over possibilities for the unseen variables.
  - This strategy is flawed as it assumes all future uncertainty will have disappeared by the time the future is reached.
  - Thus, the strategy never makes moves that seek to reveal information.
- **belief states –** games states are replaced by *possible* states along with their corresponding probabilities.
- *In games of imperfect information, it's best to reveal as little as possible, often by acting unpredictably.*