**The mini-max and alpha-beta pruning algorithms**

Mini-max algorithm2 , a recursive algorithm often used in decision-making and game theory. This algorithm generates the entire game search space for a given position (portrayable as a game tree) and returns the move associated with the highest reward irrespective of the opponent’s moves.

Mini-max can be optimized using a pruning technique called alpha-beta, where “impossible” (in the sense that neither player is interested in playing a move of this branch) branches of trees are not considered.

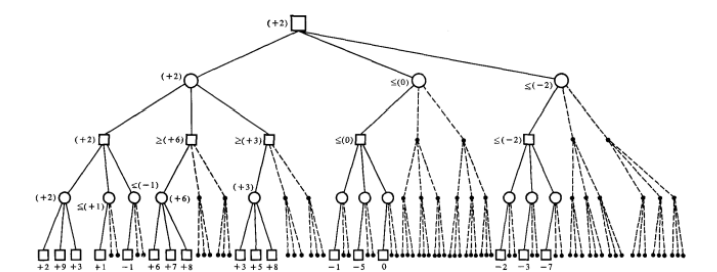


Figure: A (look-ahead) move tree in which alpha-beta pruning is fully effective if the tree is explored from left to right. Board positions for a look-ahead move by the first player are shown by squares, while board positions for the second player are shown by circles. The branches shown by dashed lines can be left unexplored without influencing the final move choice at all.

**Heuristics:**

To this day, evaluating non-trivial positions in games like checkers, chess and Go is computationally imfeasible due to the depth and width of the game tree. In order to evaluate these positions and form a decision on which move to chose next, heuristics are used. Hereby, each encountered board position gets evaluated according to certain numerical features such as the ones described in table 1 which are then combined to a linear “evaluation function”3 . Heuristics face an accuracy-effort trade-off where their simplified decision process leads to reduced accuracy [7] (in other words, heuristics face the problem of non-objectivity in their decisions), but they were the only way of evaluating board positions and thus trying to automate decision processes. In fact, the use of heuristics apes human behaviour in such board games - good players mentally jump from one line of play to another without seeming to complete any one line of reasoning. In doing so, each of these terminating board positions is evaluated according to what the player thinks about the goodness of these particular positions. If a position is deemed disadvantageous, moves leading to this position will be avoided.

# Pygame

Pygame is a multimedia library for Python for making games and multimedia applications.

It is a wrapper around the SDL (Simple DirectMedia Layer) library. In this section we indroduce the basics of pygame functions without defining classes and objects.

## Show the event loop

The most essential part of any interactive application is the ****event loop****. Reacting to events allows the user to interact with the application. Events are the things that can happen in a program, such as a

* mouse click,
* mouse movement,
* keyboard press,
* joystick action.

**Quit the event loop properly:**

In order to quit the application properly, from within the application, by using the window close button (QUIT event), we modify the event loop. First we introduce the boolean variable running and set it to True. Within the event loop we check for the QUIT event. If it occurs, we set running to False:

**Define colors**

Colors are defined as tuples of the base colors red, green and blue. This is called the RGB model. Each base color is represented as a number between 0 (minimum) and 255 (maximum) which occupies 1 byte in memory. An RGB color is thus represented as a 3-byte value. Mixing two or more colors results in new colors. A total of 16 million different colors can be represented this way