

# Pacman Homework Report

R12922A09  
Yung-Hsiang Yang

## 1. Reflex Agent

僅根據環境狀態的當前情況採取行動，而忽略過去和未來的狀態，考慮食物位置、鬼魂位置和當前情況的分數來決定下一步。它基於簡單的“condition-action”邏輯。例如，假定 pacman 當前位置，左側有 ghost，右側有 food，則 pacman 會選擇向右走。更具體地說，它使用 evaluation function 來計算當前哪一個的 action 最佳，並在可能的動作中選擇導致得分最高的狀態的動作。因為 reflex agent 可以以直覺得方式呈現，由於只需要考慮當前狀態，計算成本比其他的 agent 來的小。然而，一個明顯的缺陷是，reflex agent 可能會陷入無限循環中，可能需要採取隨機行動才能跳離循環。

## 2. Minimax Agent

Minimax 演算法，又名極小化極大演算法，是一種找出失敗的最大可能性中的最小值的演算法。找到玩家（max agent）的最佳行動，假設他的對手（min agent）也採取最佳策略。與反射代理類似，環境的每個狀態都有一個透過評估函數計算出的與之關聯的值，最大化者必須嘗試獲得盡可能的最高值，而最小化者必須嘗試做相反的事情並獲得盡可能的最低值。Minimax 應用於兩人回合製遊戲，例如井字遊戲和國際象棋，但我們可以把 minimax 作推廣，考慮到許多對抗（mini agent）的存在，以適用於更複雜的遊戲就像 pacman 一樣。

Minimax 是一種優秀的演算法，因為它可以在此類遊戲（也稱為 zero-sum game）中返回最佳的移動步驟，其中環境是完全可觀察的，遊戲不受運氣等外部因素的影響。

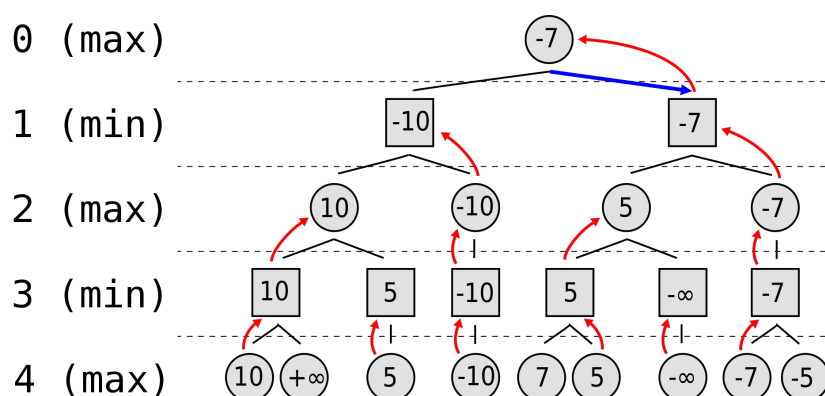


Figure 1: A minimax tree example

## 3. Alpha-Beta Pruning

Minimax 可以透過稱為 alpha-beta pruning 技巧來調整龐大的搜尋樹，這可以顯著減少計算時間。這種調整允許搜尋更快地進行，從而避免不需要搜尋的不必要的分支，因為已經存在更好

的計算結果。之所以稱為 alpha-beta pruning，是因為它需要在 minimax 函數中傳遞 2 個額外的參數，即  $\alpha$  和  $\beta$

- Alpha 是 max 目前可以保證在該等級或更高等級的最佳值，並初始化為  $-\infty$ 。
- Beta 是 min 目前可以保證在該等級或更高等級的最佳值，並初始化為  $+\infty$

這兩個新參數用於修剪那些不可能影響最終決策的分支。每當 min 玩家 ( $\beta$ ) 保證的最大分數小於 max 玩家 ( $\alpha$ ) 保證的最小分數時，因此如果  $\beta < \alpha$ ，則 max 玩家不需要考慮進一步的後代節點，因為它們的分數不會比現在的結果來的好，自然就不會被訪問到。

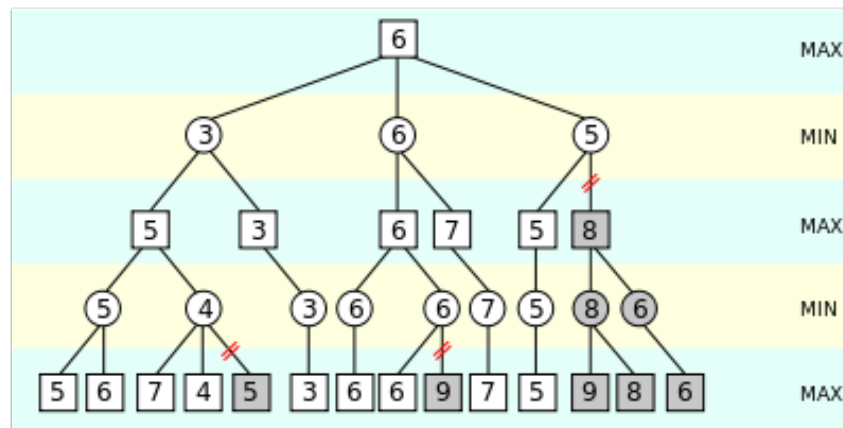


Figure 2: A minimax tree example