

電腦科學邏輯基礎期末報告

107703008 資科三 蔡桂毓

1. 基本的搜尋架構

這次我所使用的搜尋架構是較基本的 2 步驟 minimax-search，從一開始 step1 下完之後判斷去分析再進行 step2，分析完並給出最終答案。

```
step1(X, Y),
(
  addMove(Player, X, Y),
  (
    fourtowin(Opponent, _, _)
  -> (undoMove(Player, X, Y))
  ; ( (three(Opponent, _, _, 0, 0), \+fourtowin(Player, _, _))
    -> (
      forall((three(Opponent, L, A, 0, 0), member([X1, Y1], A)), (step1(X1, Y1)
      -> true
      addMove(step2, X1, Y1))),
      forall(step2(X1, Y1),
        (
          addMove(Opponent, X1, Y1),
          (
            winSecure(Opponent)
          -> undoMove(Opponent, X1, Y1), false
          ; true),
          undoMove(Opponent, X1, Y1)
        )
      )
    )
  )
)
```

2. 評估盤面的方式

由於此為較基本的搜尋，所以分別對 2 步驟進行分析。在進行第一步前，先判斷是否有 4 子可以連線並致勝。再來是在第一步後分析此落點是否位在對家 4 子準備落子並取勝的點，若無則判斷是否為自己活 3 子的落點，落子後位置是否能為自己創造出 4 子局面。第二步的分析如同第一步，若中間有任何不符則退回上一步，並切換落點再進行重新分析。

3. 速度優化

因為在進行判斷時發現自己所做的落點判斷條件，對於黑子的各項禁手皆無法順利完成，故參考同學所使用的方法來判斷是否為禁手。但在聽懂同學所跟我解釋的判斷方法並實行後速度卻非常緩慢，僅在一個落點的判斷需求會超過對於比賽條件之反應時間 5 秒，在修改無果之後只好轉而將成是主要作為防守並移除對禁手的判斷來做速度優化。

```
%black only
doubleDead4(Player, X, Y) :-
    addMove(Player, X, Y),
    (    fourtowin(Player, L1, _),
      fourtowin(Player, L2, _),
      L1 \= L2,
      member([X, Y], L1),
      member([X, Y], L2))
    -> undoMove(Player, X, Y)
    ; undoMove(Player, X, Y),
      false.

doubleLive3(Player, X, Y) :-
    addMove(Player, X, Y),
    (    live3(Player, L1, _),
      live3(Player, L2, _),
      L1 \= L2,
      member([X, Y], L1),
      member([X, Y], L2))
    -> undoMove(Player, X, Y)
    ; undoMove(Player, X, Y),
      false.

triggerLongLine(Player, X, Y) :-
    addMove(Player, X, Y),
    (    longLine(Player, L1),
      member([X, Y], L1))
    -> undoMove(Player, X, Y)
    ; undoMove(Player, X, Y),
      false.
```

雙死四：

判斷是由下一步棋後判斷黑子是否具有 2 條不一樣的 4 子，若有則移除此步棋並回傳 false，代表下到禁手；反之則移除此步棋並回傳 true，代表此步不為禁手。

雙活三：

判斷是由下一步棋後判斷黑子是否具有 2 條不一樣的 3 子，無論是連 3 或跳 3 都為一樣的情形作判斷。若有則移除此步棋並回傳 false，代表下到禁手；反之則移除此步棋並回傳 true，代表此步不為禁手。

長連：

判斷是由下一步棋後判斷黑子是否會產生超過 5 子的勝利，及判斷落子之 4 垂直、水平、正斜及逆斜之反方向是否有超過 5 子之判斷，若有則移除此步棋並回傳 false，代表下到禁手；反之則移除此步棋並回傳 true，代表此步不為禁手。

4. 其他心得

在這一學期的課程中，對於 prolog 這門程式語言是又愛又恨。在一開始學習這樣的新語言時，也恰巧這學期也接觸了許多沒接觸過甚至也都沒聽過的語言，而我也對於這樣的挑戰充滿期待。在第一次作業做 $3n+1$ 時，對於 prolog 有許多不懂之處，並從老師的課堂中、投影片上及網路的資源來進行作業，對於 is 跟 = 的用法等等的邏輯與句也多了新的認識，雖然是簡單的作業但也是花了一個晚上的時間來完成，但也因為是新的語言，所以這也讓我獲得不小成就感。但到期中考時對於程式邏輯的輕忽，導致考試沒有注意到一些非常基本的地方並獲取了相對低的分數，而後決定要多用更多心思在這門課中。不巧的是又逢疫情爆發，許多之前能在班上詢問的朋友都紛紛返鄉。但在這次的作業難度卻也遠遠超出我的想像，雖然自己第二次作業邏輯之島的部分已有提早開始下筆準備，而我在撰寫程式時也詢問朋友非常多次但我都覺得自己的程式邏輯沒有問題，在瘋狂詢問及反覆測試了三天才終於找到一個小地方的錯誤讓我這次能順利交出作業，但這也讓我對於 prolog 是更多得不了解。在期末報告時一開始是非常緊張也沒什麼頭緒，上網查資料也沒什麼想法能撰寫。直到老師公佈說有 tic tac toe 的方法之後我便試著將其改成五子棋，但總也有許多不一樣之處導致我無法得到我想要的輸出也導致了我第一次程式只能碰運氣般猜猜看能不能獲得分數，最後只好仍然打擾同學來問說應該要如何做比較好，故也在最後趕工之下完成這次的程式。這學期的課程在疫情下有些無所適

從，自己也知道這樣的分數應該不會太漂亮，但仍然謝謝老師這學期對我的指導以及鼓勵，在和老師通話談論之前甚至有想過乾脆直接放棄完成程式，但在老師提點後讓我決定就算做不好也應該要盡力完成。故最後的一個禮拜在做完其餘報告之後就專心投入完成程式，結果可能不甚理想但也是有做出最後的成果，完成這次的課題。雖然比較可惜的是之後沒辦法再挑戰一次老師的這門課，但仍然謝謝老師和助教這學期辛苦的教導。