Computation, Information, and Intelligence (COMS/ENGRI/INFO/COGST 172), Fall 2005
9/9/05: **Lecture aid – Bounded exploration and pruning**

Agenda: More observations about the game-playing set-up. Coping with large game trees using pruning.
Announcements: Reminder: Prof. Lee will not be holding her regular office hours on Monday 9/12. (She will be out of town. There will still be lecture, and all the other office hours for the week will be held.)

I. Example evaluation functions

1. Problem: begin and maintain a stock portfolio for the month with the aim of making a profit. States: portfolio and the prices of each item of the portfolio.

   Function: the value of the portfolio.

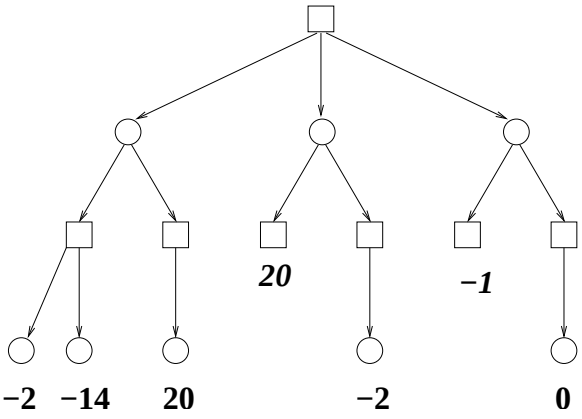
2. Problem: win a game of checkers. States: legal board positions.

   Function:

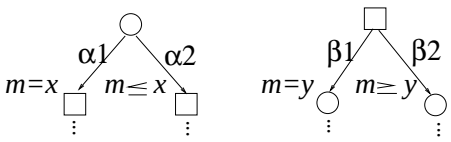
$$f(s) = \begin{cases} +100 & \text{s is a state in which you have won} \\ -100 & \text{s is a state in which you have lost} \\ n_1 - n_2 & \text{otherwise} \end{cases}$$

   where n_1 and n_2 are the number of pieces you and your opponent, respectively, have in the corresponding board position, with kings for counting double.

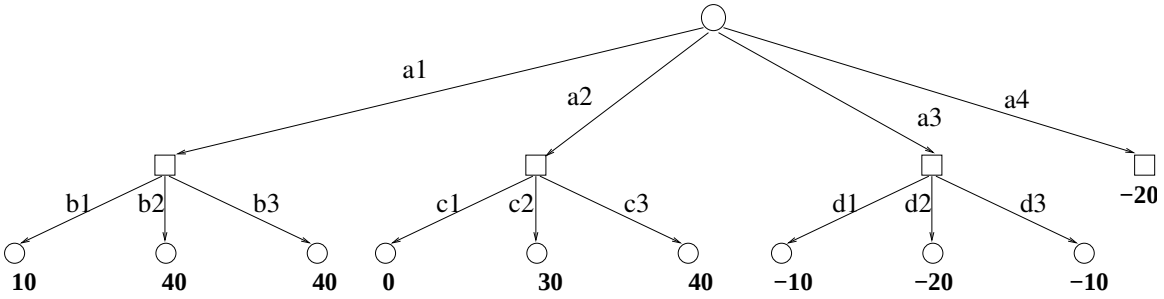
II. Another game tree, with leaves at different depths ◯ and □ indicate player 1's and player 2's turn to move, respectively.



III. Pruning The fundamental principle is that we need not examine nodes whose (pseudo-)minimax values can't affect decisions "higher up" in the game tree. Two canonical situations are the following, where ◯ and □ indicate player 1's and player 2's turn to move, respectively, and by each node we have indicated (using variables x and y) our current knowledge about its minimax value m:



IV. Pruning example Here is what is either a full (small) game tree or the portion that would be explored using bounded exploration without any pruning (α, β, γ, and δ (alpha, beta, gamma, and delta) are the first four (lower-case) letters of the Greek alphabet). The idea is to not have to look at all of its nodes.



Notation: Circle leaf-node (pseudo-)minimax values that are consulted. Use hash marks to indicate pruning sites, below which we don't bother searching. Write current constraints by nodes, and cross out out-dated constraints.