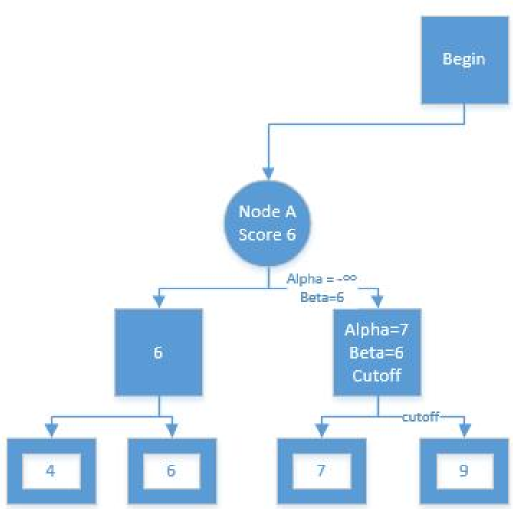
Homework 4

李云志 2015K8009929014

2017 年 9 月 23 日

Theorem 1 *Negascout is correct*
Basic idea: most moves after the first will result in cutoffs, so we try to prove them inferior by using a minimal alpha-beta window.

Proof 1 *First step: Alpha-Beta search is correct.*
Alpha=best already explored option along path to the root for maximizer (square)
Beta=best already explored option along path to the root for minimizer (circle)
Every step, parent node will pass the value of alpha and beta to its child node and update its score, the value of alpha and beta when it get the return value from its child node.
When, alpha$\geq$ beta, then we get a cutoff.
Because, the value of alpha means that the maximizer can get a score higher or equal than alpha, the value of beta means that the minimizer can limit a score equal or less than beta,so if alpha$\geq$ beta, the ideal minimizer will definitely not choose this branch, so it si no need to keep searching.
For example, in Graph 1, Node A update its score from $-\infty$ to 6 and pass values of alpha and beta to its right child. When this right child update its score and alpha's value, we get a cutoff on its right branch, because alpha $\geq$ beta.



Proof 2 *Negascout search is correct*

The difference between Alpha-Beta search and Negascout is the using of the minimal window, which means in every step (except the first action) the window is revised to (alpha, alpha+1).

It is equal to Alpha-Beta search because when the value of alpha is updated to a, it is just like that there is a minimizer which have limited the score to equal or less than a+1, so we can use window (alpha,apha+1) to keep searching.

However, Negascout sometimes is better than Alpha-Beta because it can cut some branches that Alpha-Beta search must search precisely.

For example, in Graph 2, if we use Alpha-Beta search, the window begin with value(6,+∞), so we must go down to node B and search all its child.

When we use Negascout, the branch from A to B will be cut off immediately when the search arrive at node c, because the updated value alpha (9) is larger than beta(7).