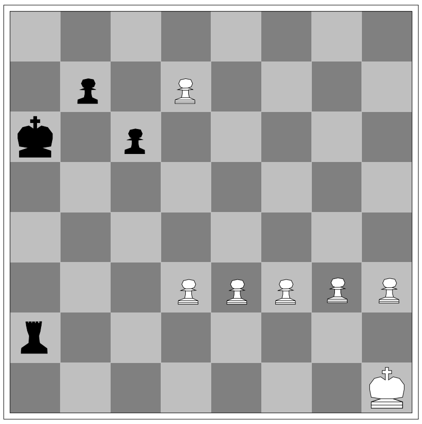

Figure 5.5 The horizon problem. A series of checks by the black rook forces the inevitable queening move by white "over the horizon" and makes this position look like a slight advantage for black, when it is really a sure win for white.

PRUNING

ALPHA-BETA PRUNING

without examining it is called **pruning** the search tree. The particular technique we will examine is called **alpha-beta pruning**. When applied to a standard minimax tree, it returns the same move as minimax would, but prunes away branches that cannot possibly influence the final decision.

Consider the two-ply game tree from Figure 5.2, shown again in Figure 5.6. The search proceeds as before: A_1, then A_{11}, A_{12}, A_{13}, and the node under A_1 gets minimax value 3. Now we follow A_2, and A_{21}, which has value 2. At this point, we realize that if MAX plays A_2, MIN has the option of reaching a position worth 2, and some other options besides. Therefore, we can say already that move A_2 is worth *at most 2* to MAX. Because we already know that move A_1 is worth 3, there is no point at looking further under A_2. In other words, we can prune the search tree at this point and be confident that the pruning will have no effect on the outcome.

The general principle is this. Consider a node *n* somewhere in the tree (see Figure 5.7), such that Player has a choice of moving to that node. If Player has a better choice *m* either at the parent node of *n*, or at any choice point further up, then *n will never be reached in actual play.* So once we have found out enough about *n* (by examining some of its descendants) to reach this conclusion, we can prune it.

Remember that minimax search is depth-first, so at any one time we just have to consider the nodes along a single path in the tree. Let α be the value of the best choice we have found so far at any choice point along the path for MAX, and β be the value of the best (i.e., lowest-value) choice we have found so far at any choice point along the path for MIN. Alpha-beta search updates the value of α and β as it goes along, and prunes a subtree (i.e., terminates the recursive call) as soon as it is known to be worse than the current α or β value.

The algorithm description in Figure 5.8 is divided into a MAX-VALUE function and a MIN-VALUE function. These apply to MAX nodes and MIN nodes, respectively, but each does the same thing: return the minimax value of the node, except for nodes that are to be pruned (in