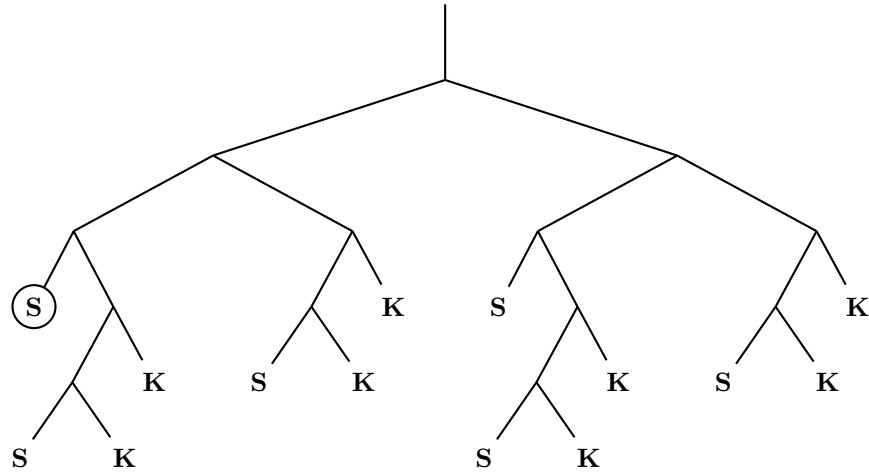
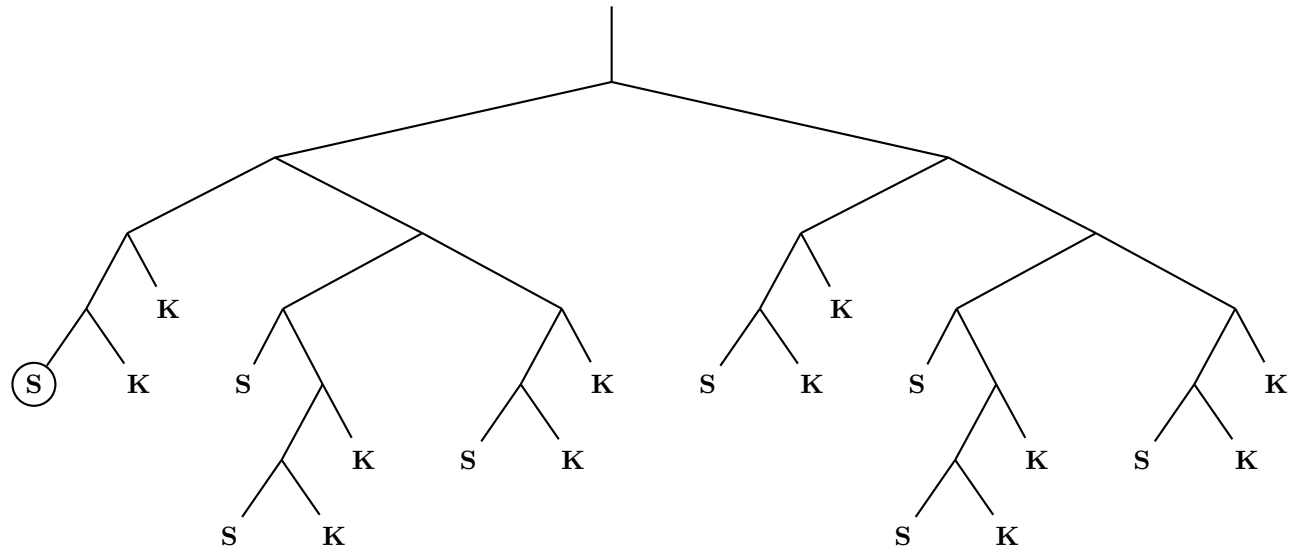


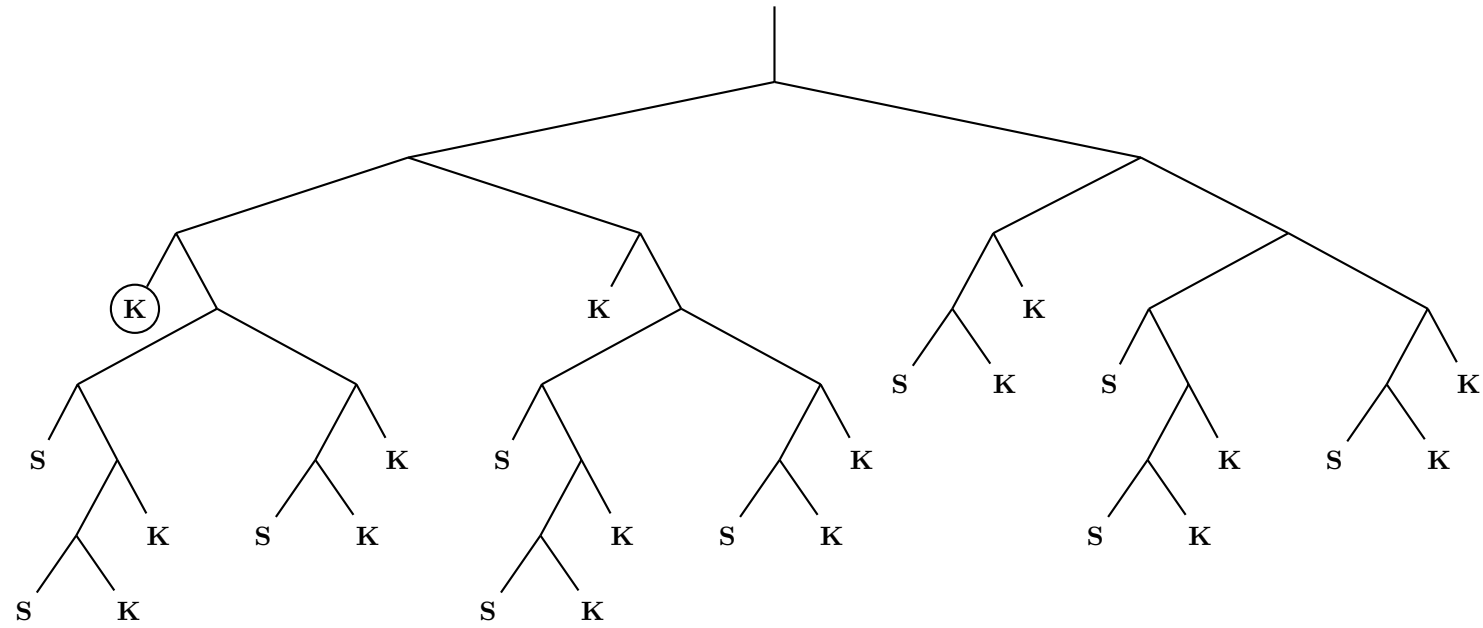
Step 1: **S(SKK)(SKK)(S(SKK)(SKK))**



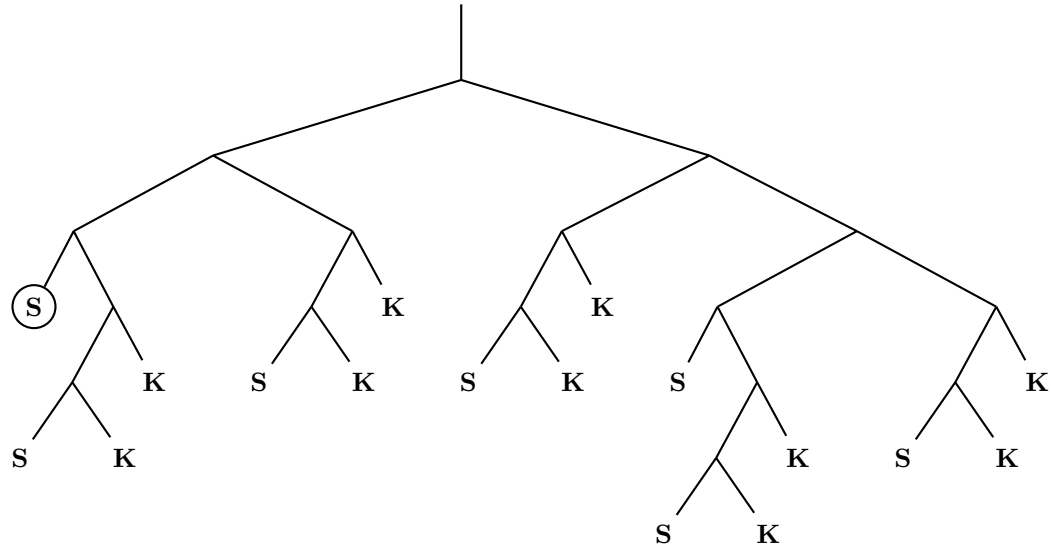
Step 2: **SKK(S(SKK)(SKK))(SKK(S(SKK)(SKK)))**



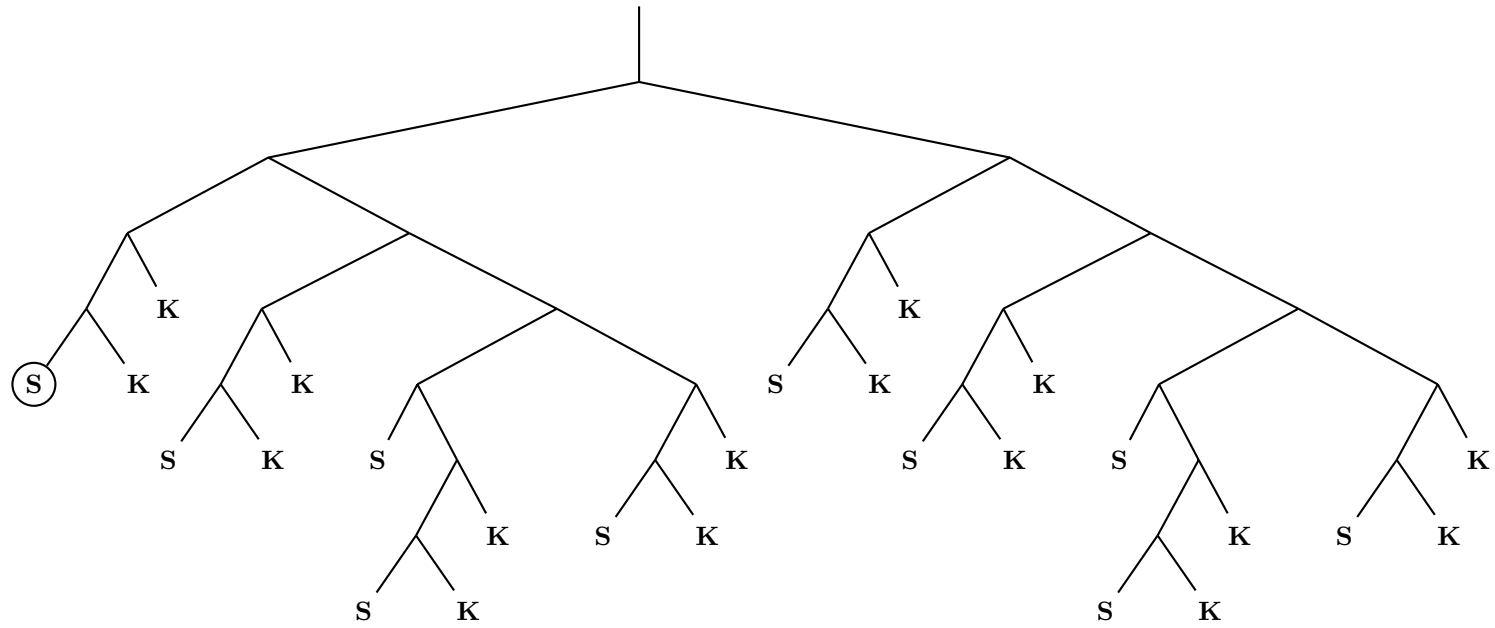
Step 3: **K(S(SKK)(SKK))(K(S(SKK)(SKK)))(SKK(S(SKK)(SKK)))**



Step 4:  $S(SKK)(SKK)(SKK(S(SKK)(SKK)))$

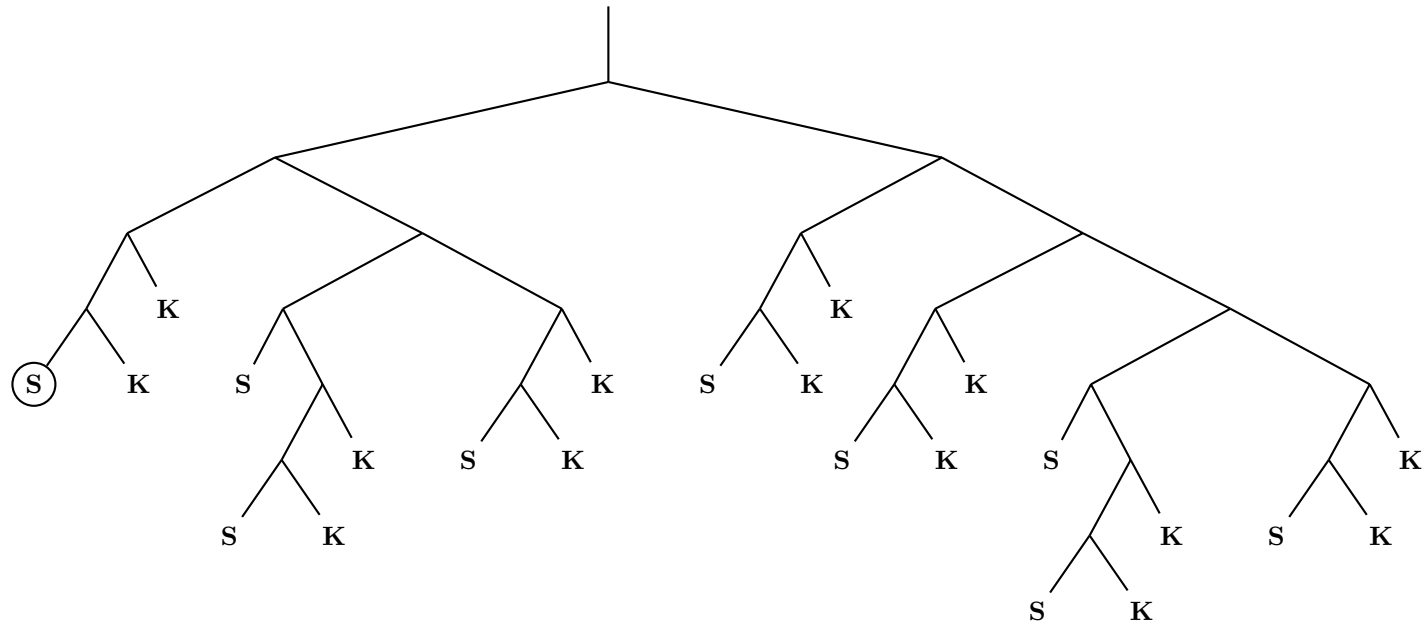


Step 5: **SKK(SKK(S(SKK)(SKK)))(SKK(SKK(S(SKK)(SKK))))**

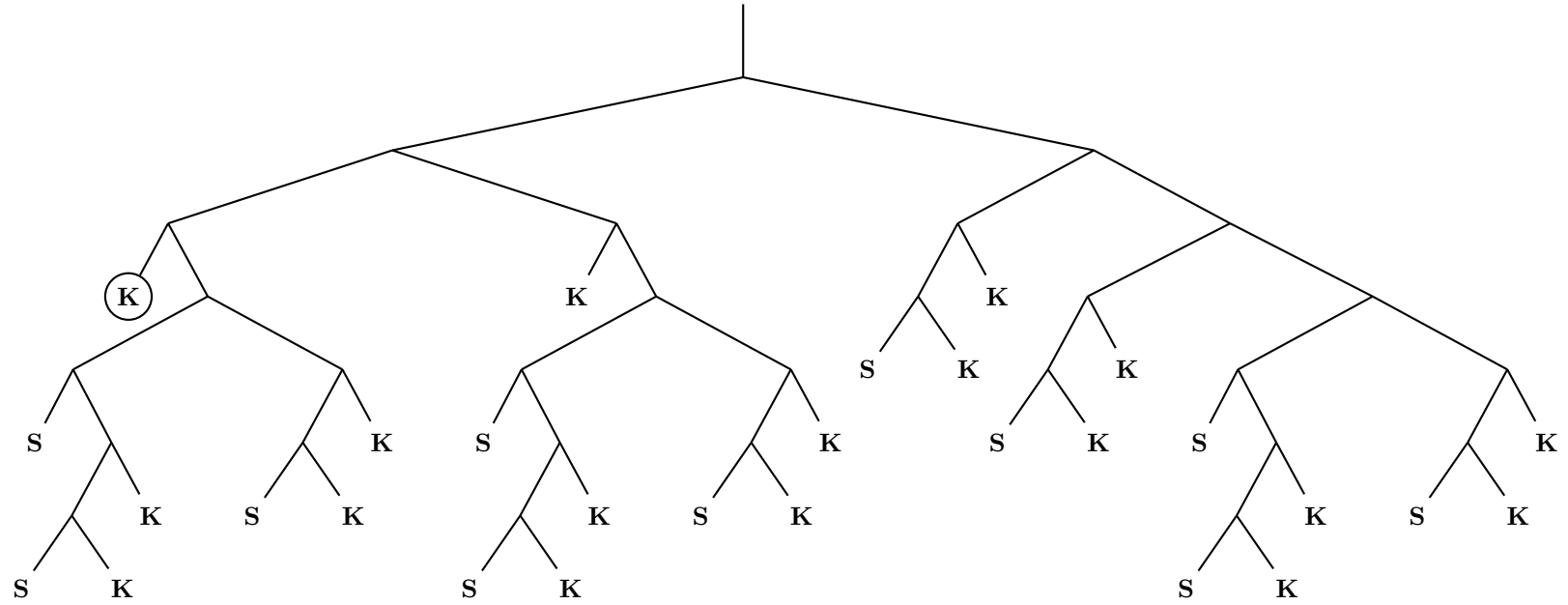


The diagram illustrates a binary tree structure, likely representing a search space for a 3x3 grid. The root node branches into two main paths. The left path leads to a node labeled 'K' (circled), which then branches into 'S' and 'K' nodes. The right path leads to a node labeled 'K', which branches into 'S' and 'K' nodes. The tree continues to branch down to a third level, with nodes labeled 'S' and 'K'.

Step 7: **SKK(S(SKK)(SKK))(SKK(SKK(S(SKK)(SKK))))**

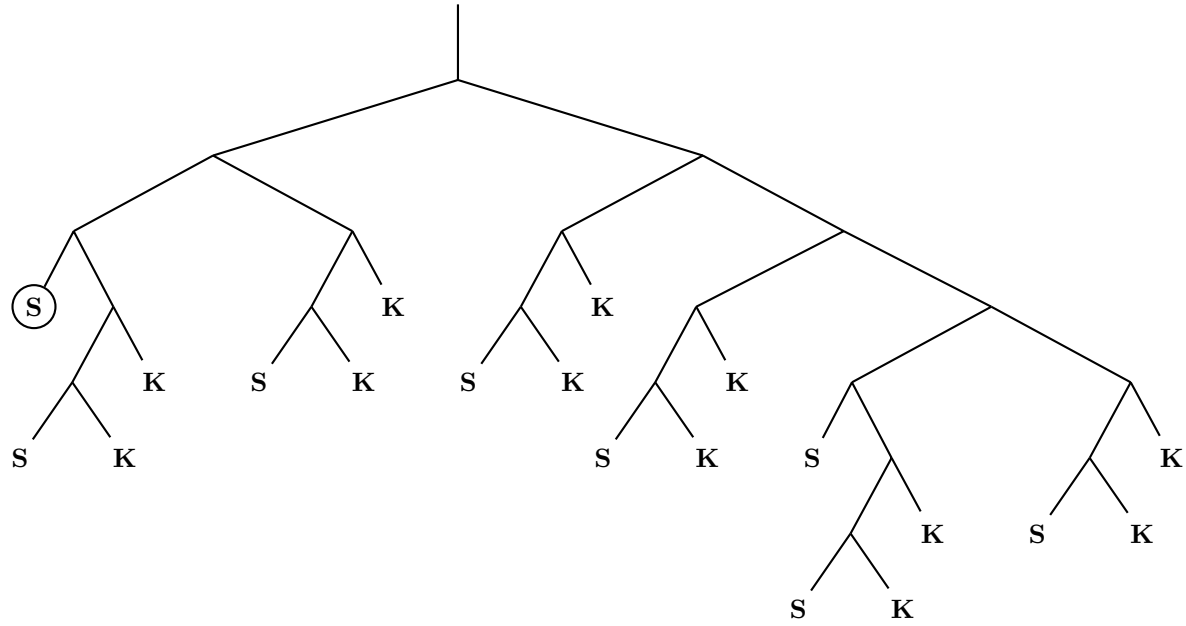


Step 8: **K(S(SKK)(SKK))(K(S(SKK)(SKK)))(SKK(SKK(S(SKK)(SKK))))**

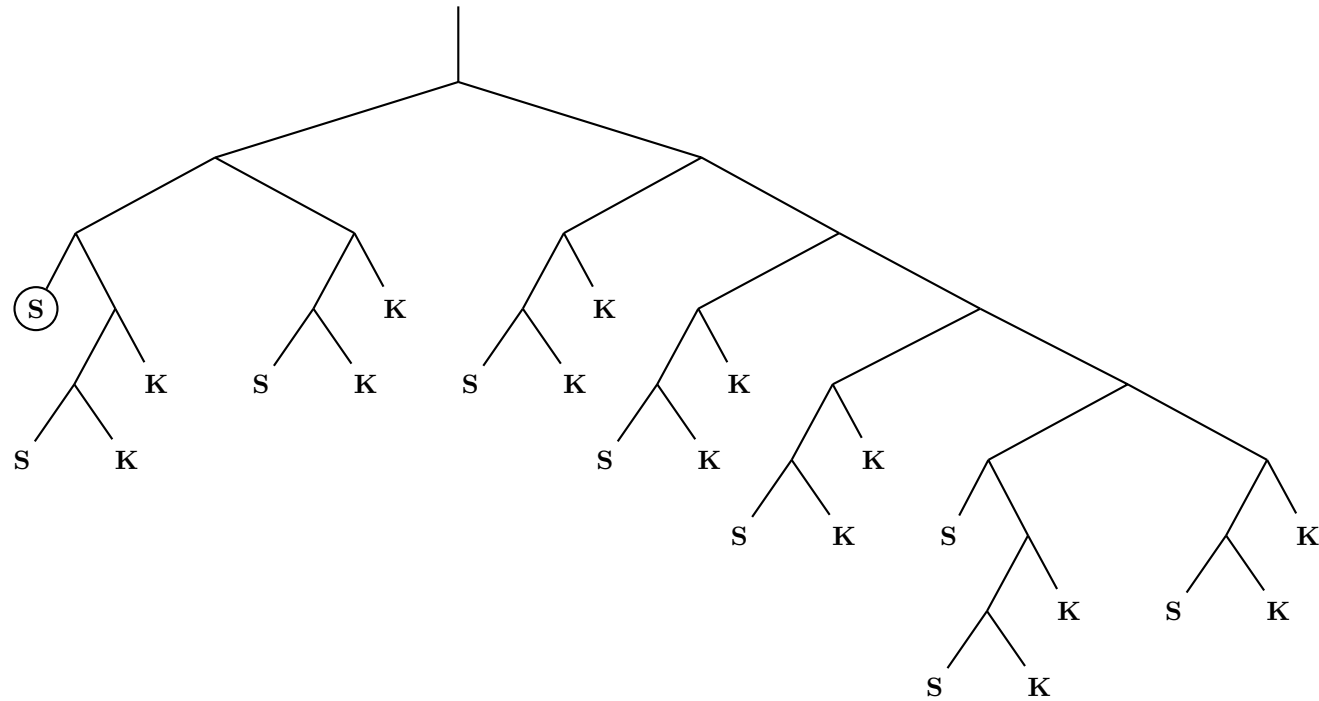




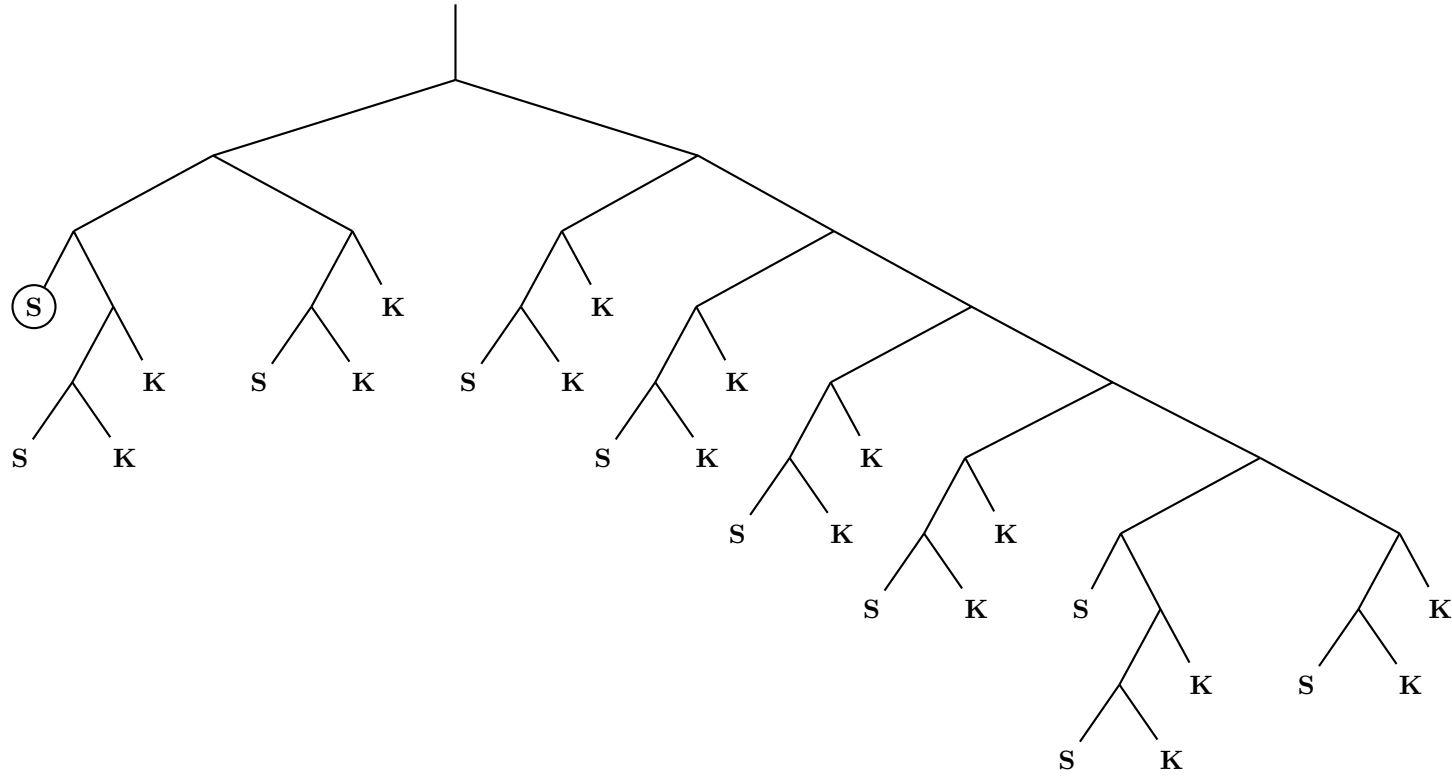
Step 9:  $S(SKK)(SKK)(SKK(SKK(S(SKK)(SKK))))$



Step 16:  $S(SKK)(SKK)(SKK(SKK(S(SKK)(SKK))))$



Step 25:  $S(SKK)(SKK)(SKK(SKK(SKK(S(SKK)(SKK))))))$



Step 36:  $S(SKK)(SKK)(SKK(SKK(SKK(SKK(S(KK)(SKK))))))$

