**Local Search in CSP**

**Procedure:**

1. **Define the Problem**:

* List the variables and their respective domains using variable/2.
* Define constraints between variables using constraint/2.

2. **Check Valid Assignments**:

* Use the valid\_assignment/1 predicate to ensure the constraints are satisfied for a given set of variable assignments.

3. **Search for a Solution**:

* The local\_search/1 predicate performs the search:
  + Iterate over the domain values of the variables.
  + Combine the values into an assignment.
  + Check if the assignment satisfies the constraints.

4. **Output the Solution**:

* When a valid assignment is found (i.e., satisfies all constraints), return it as the solution.

**Alpha Beta Pruning**

**Procedure**

1. Define terminal node values: Use value/2 to assign static evaluation scores to terminal nodes.
2. Define tree structure: Use move/2 to represent parent-child relationships in the game tree.
3. Alpha-Beta pruning: Implement the alphabeta/5 predicate to recursively traverse the tree while maintaining Alpha and Beta bounds to optimize the search.
4. Evaluate child nodes: Use the evaluate/6 predicate to process children, pruning branches when Alpha exceeds Beta.
5. Run the query: Execute a query like ?- alphabeta(root, Depth, -inf, inf, BestValue). to compute the best value of the root node based on the tree and depth.