

# CS404 - HW2

Tahir Turgut - 27035

## 1 CSP Representation of Hashi Puzzle

In an  $N \times N$  Hashi puzzle, each cell represents either a possible bridge or a node. Nodes can be counted as constants, since they are immutable. On the other hand, all other edges (cell intervals) which contains a vertical or horizontal path to one node to another can be counted as a variable which has domain of  $[0, 1, 2]$ , corresponding the bridge count that is currently belonging to that cell. Also, constraints can be represented as following:

- (i) Every bridge starts from a node and ends in a node as a straight line. (If the bridge is vertical, there must be a *node* in that column. If the bridge is horizontal, there must be a *node* in that row.) -This is asserted via variables
- (ii) Edges that contains a node (island) has to sum up to value of that constant.
- (iii) There cannot be a vertical bridge in the upper or lower cell of a cell that contains horizontal bridge.
- (iv) There cannot be a horizontal bridge in the right or left cell of a cell that contains vertical bridge.

## 2 Which one is more appropriate: A\* or CSP?

A\* algorithm will keep the being of the visited states as a checkpoint, in case of a failure in satisfying the constraints. It will try to assign a value on that state; if it is successful, algorithm will keep expanding on that newly created state. On the other hand, CSP algorithm will try to assign values to the variables which are unassigned still and will not assign if there is a constraint failure. Furthermore, it will not keep the visited states or it has no frontier queue. Therefore, in the A\* algorithm, the space consumption will be enormously larger than CSP.

In addition to those, thanks to additional improvements such as Forward Checking, Arc Consistency etc., it is possible to detect a state will lead to failure before actually creating that specific state in CSP, whereas A\* will expand a node that will lead to a violation from frontier queue if it is the state will less cost -predicted-.

Even though negation of violation count heuristic helps A\* to avoid from making worse moves, it stores and expands more nodes than CSP. It makes, CSP more advantageous in terms of space and time complexity.