Constraint Satisfaction Problems

A solution is a state in which the domain fits the constraints

Initial state = {}

Successor function: assign a value too an unassigned variable that does not conflict with current assignment; fail if no legal assignments

Goal test: the current assignment is complete

1. Same for a cps
2. Every solution at depth n with n variables
3. Minimum remaining values: expand nodes with only one choice first

8 Queens

Variables: columns x8

Domain: rows x8

Dictionary of [c:[r1,r2…], c2…]

State = vector of sets that starts out as [{1-8},{1-8}…{…}]

So if you choose c1 = 3, remove 3 from every set, and then diagonals 🡪 c2 = {5,3},

Constraints:

* (col cannot be the same) ci != cj 🡪 8C2
* Abs(ci – cj) != abs(i – j) 🡪 8C2

While (true)

If fringe empty then fail

H = fringe.pop()

If n is goal, return n

Pick an unassigned column Ci in n

Pick a value, r, for ci

Remove stuff above

Add new state to fringe