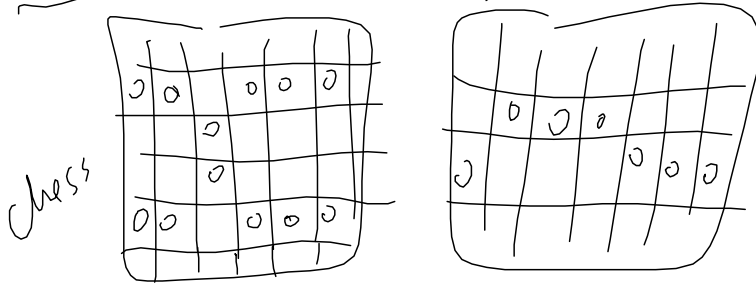


## Search 2

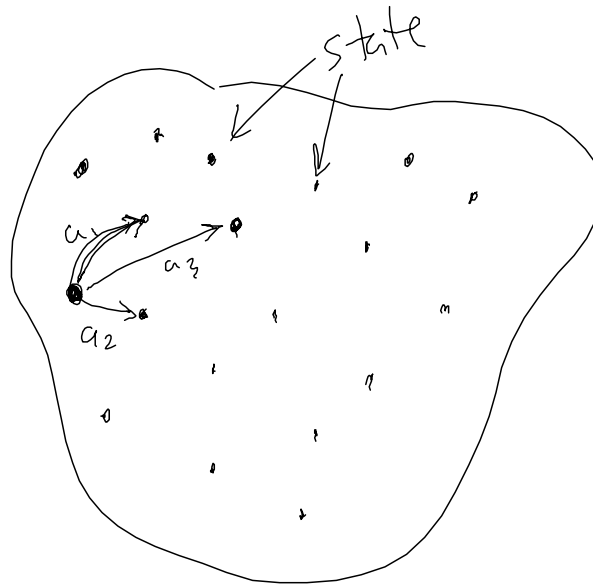
Friday, September 2, 2016 8:58 AM

State: unique config of world.



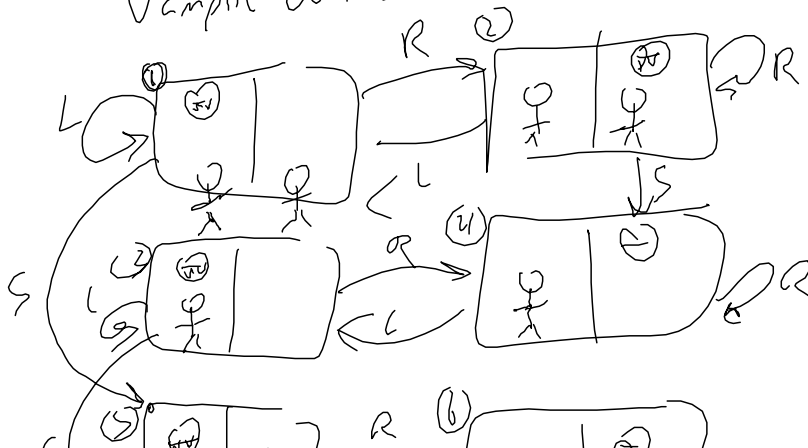
Vampires: { location: { 1, 2 }  
 1 empty: T/F  
 2 empty: T/F }

State space



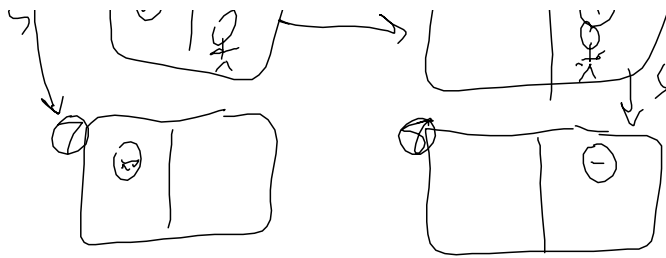
Vampire World

Actions: { L, R, S }



Solution:

{ S, R, S }



8-Puzzle

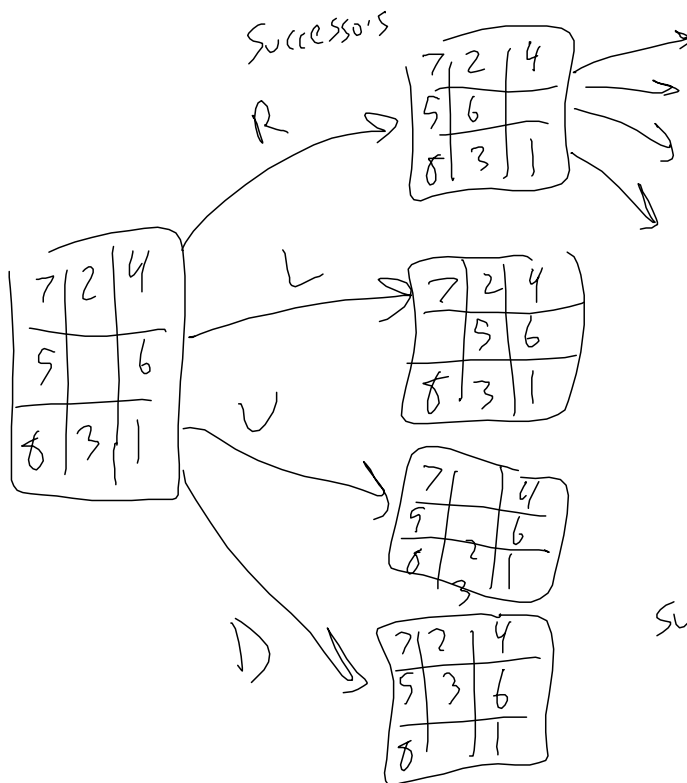
Goal

init

7	2	4
5		6
8	3	1

1	2	3
4	5	6
7	8	

Actions:

 $\{R, L, U, D\}$ 

SUCCESSOR Function:  
generates new states  
from old states

$$\text{succ}(s, A) \rightarrow \{s'_1, s'_2, \dots, s'_n\}$$

$$A = \{R, L, D, U\}$$

For each  $a \in A$  do:

$s' \leftarrow \text{apply } a \text{ to } s$

- Evaluation of algorithms
- ① Completeness - Does it always find a solution?  $\leftarrow$   
 - Does it visit all states?  $\leftarrow$   
 it will find a goal

② Time Complexity - # of states generated  
- worst, best, average

③ Space complexity - # states stored in memory at any given time.

④ Optimality - Does it always find the "lowest-cost" (shortest) solution?

## UNINFORMED SEARCH

\* Fully Observable - "map" in our heads.

\* Search alg's "imagine movement"

\* solves problem in its head before doing any execution.

## Random Search

- given init state
- pick a legal act
- Generate successor
- update current

