COMP 472: Artificial Intelligence
State Space Search part #3
State Space Representation video #1

- Russell & Norvig – Sections 3.1-3.3

Today

1. State Space Representation

video #1

2. State Space Search
   a) Overview
   b) Uninformed search
      1. Breadth-first and Depth-first
      2. Depth-limited Search
      3. Iterative Deepening
      4. Uniform Cost
   c) Informed search
      1. Intro to Heuristics
      2. Hill climbing
      3. Greedy Best-First Search
      4. Algorithms A & A*
      5. More on Heuristics
   d) Summary

Motivation

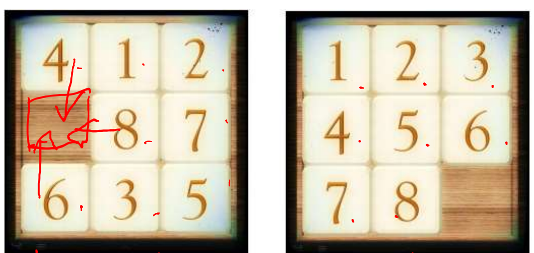
initial

goal

1970

我们不知道具体怎样从initial state变成goal state，但我们可以通过一些action，从一个state到另一个state，把这些possible state列出来，搜索goal state

- Many AI problems, can be expressed in terms of going from an initial state to a goal state
  - *Ex: to solve a puzzle, to drive from home to Concordia…*



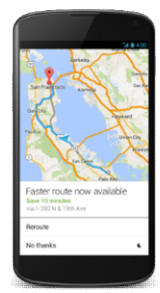
initial goal

8-puzzle

15 moves

200 moves

15move
200move不一定



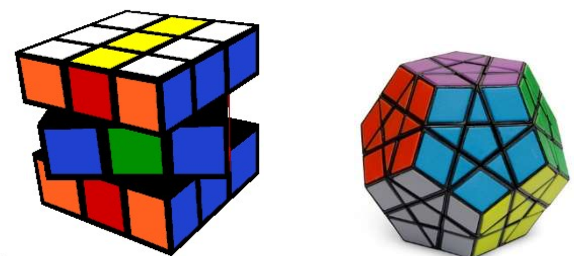
Google Maps

home

left right

street right left

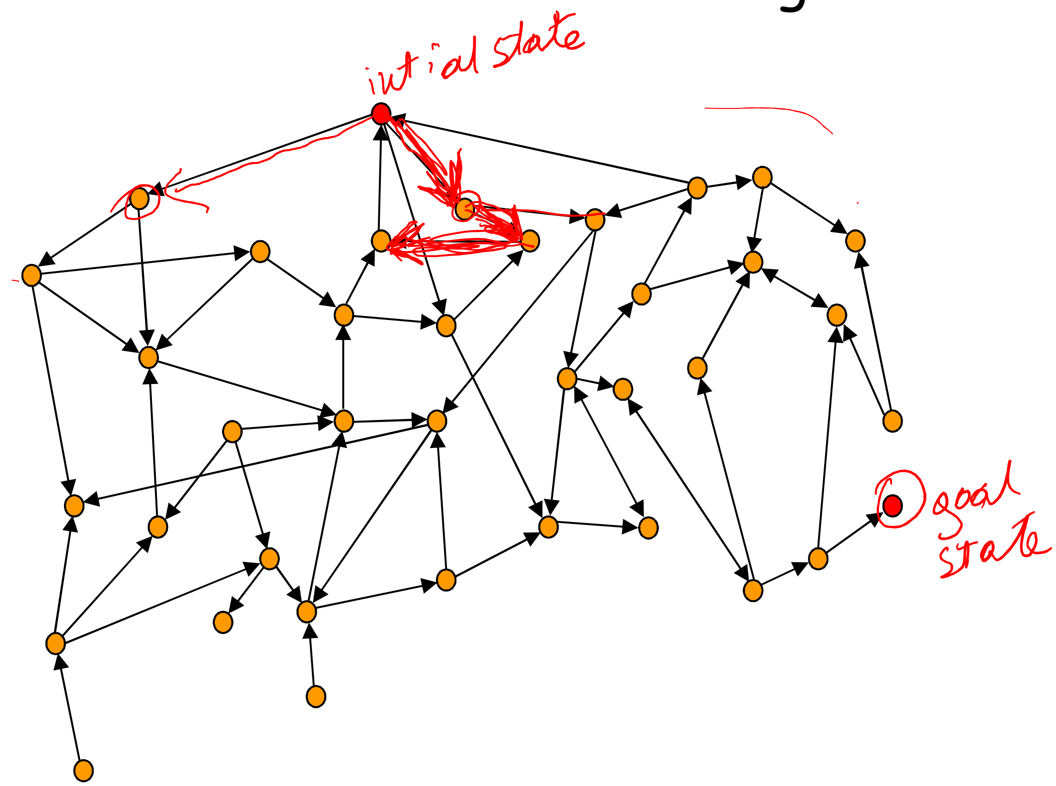
Concordia



Rubik's cube

Motivation

- Often, there is no direct way to find a solution to go from the initial state to a goal state
- but we can list the possibilities and search through them



Example: 8-Puzzle

State: Any arrangement of 8 numbered tiles and an empty tile on a 3x3 board

8	2	empty
3	4	7
5	1	6

Initial state

1	2	3
4	5	6
7	8	

Goal state

there are several standard goals states for the 8-puzzle

1	2	3
4	5	6
7	8	

1	2	3
8		4
7	6	5

...

$(n^2\text{-}1)$-puzzle

8	2	
3	4	7
5	1	6

8-puzzle

4

4

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

15-puzzle

....

15-Puzzle

Invented in 1874 by Noyes Palmer Chapman …
but Sam Loyd claimed he invented it!

Sam Loyd

see Sam Loyd's book of puzzles:
https://archive.org/stream/CyclopediaOfPuzzlesLoyd/Cyclopedia_of_Puzzles_Loyd#mode/2up

https://en.wikipedia.org/wiki/Sam_Loyd

State Space

- Problem is represented by:
  1. Initial State
     - starting state
     - ex. unsolved puzzle, being at home
  2. Set of operators / moves / actions
     - actions responsible for transition between states
  3. Goal test function
     - Applied to a state to determine if it is a goal state
     - ex. solved puzzle, being at Concordia
  4. Path cost function
     - Assigns a cost to a path to tell if a path is preferable to another

如果我们能用这四步描述问题，那么就能使用state space search

- State space:
  - the set of all states that can be reached from the initial state by any sequence of action
- Search algorithm:
  - how the search space is visited

最后我们需要一个search algorithm

	cost	cost
up	1	3
down	1	1
left	1	2
right	1	4
	least nb of moves	lowest cost

有的时候求最低cost

有的时候求最小move数量(COST一样)

goal(···) → T / F

goal(···) = T

initial

goal

comp 352

Example: The 8-puzzle

goal

8	2	
3	4	7
5	1	6

= false

1 Initial state

goal

1	2	3
4	5	6
7	8	

= True

Goal state

2 - Set of operators:

blank moves up, blank moves down, blank moves left, blank moves right

格子例如7往上移，相当于空格往下移，我们移动空格

3 - Goal test function:

state matches the goal state

4 - Path cost function:

each movement costs 1

so the path cost is the length of the path (the number of moves)

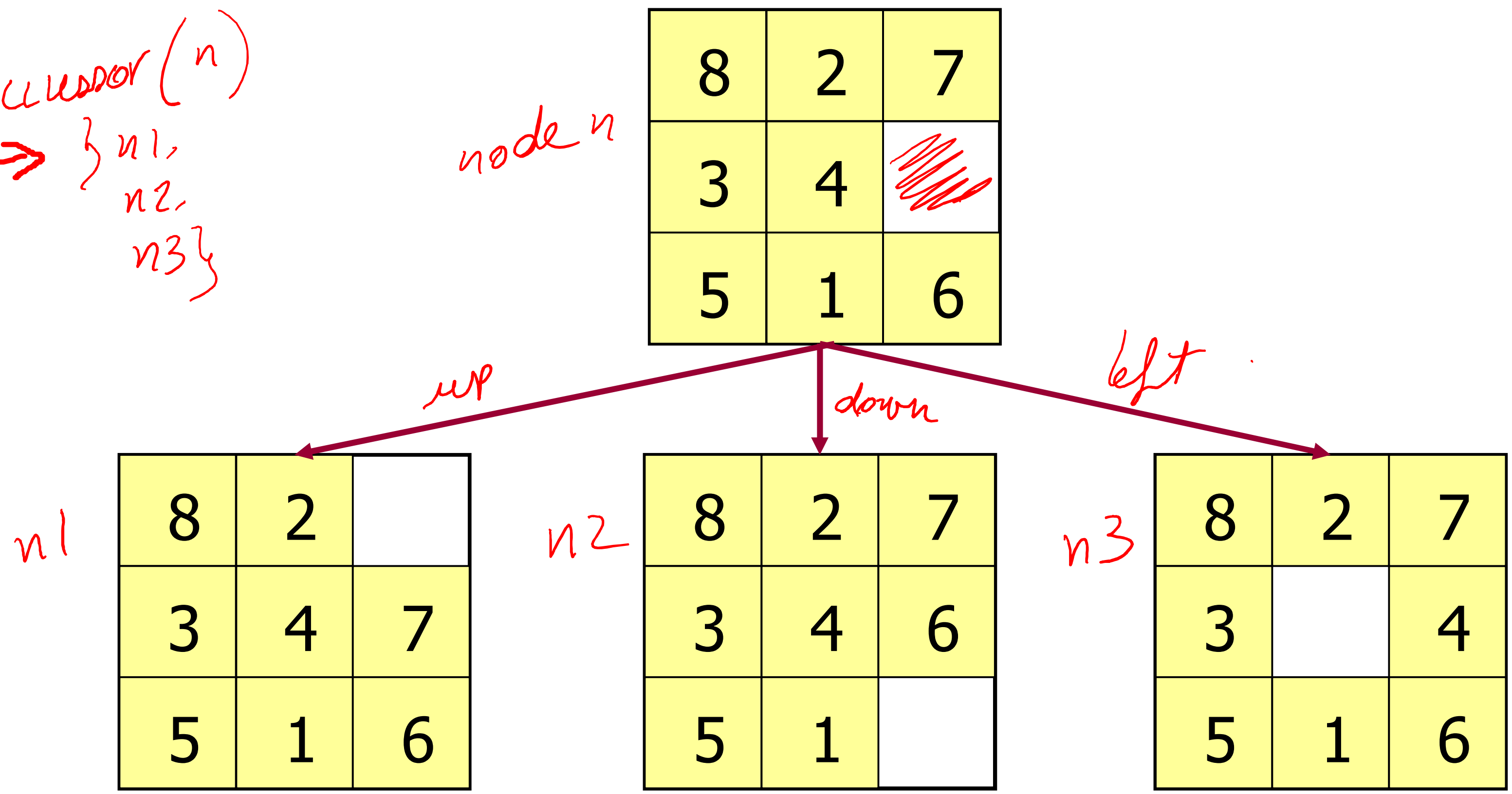
source: G. Luger (2005)

8-Puzzle: Successor Function 后继函数

{children

successor(n)生成了n1,n2,n3

successor(n)
→ {n1,
n2,
n3}



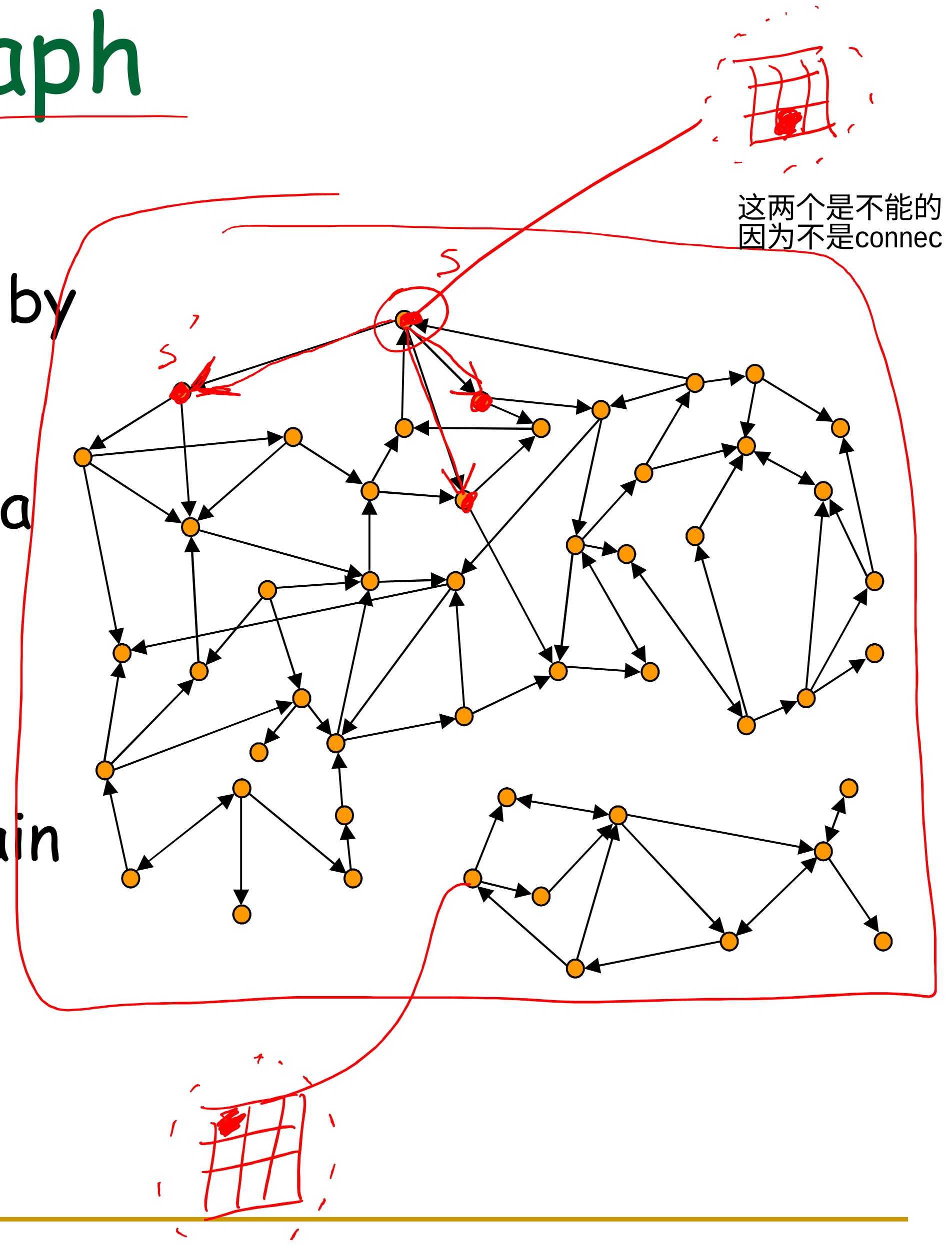
Search is about the exploration of alternatives

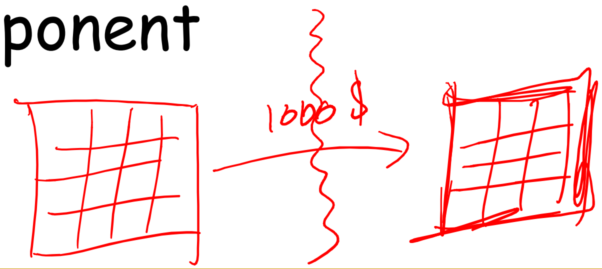
State Space Graph

对每个node使用successor function
如果结果是其中一个点，相连

- Each state is represented by a distinct node
- An arc (or edge) connects a node s to a node s' if s' $\in$ SUCCESSOR(s)
- The state graph may contain more than one connected component

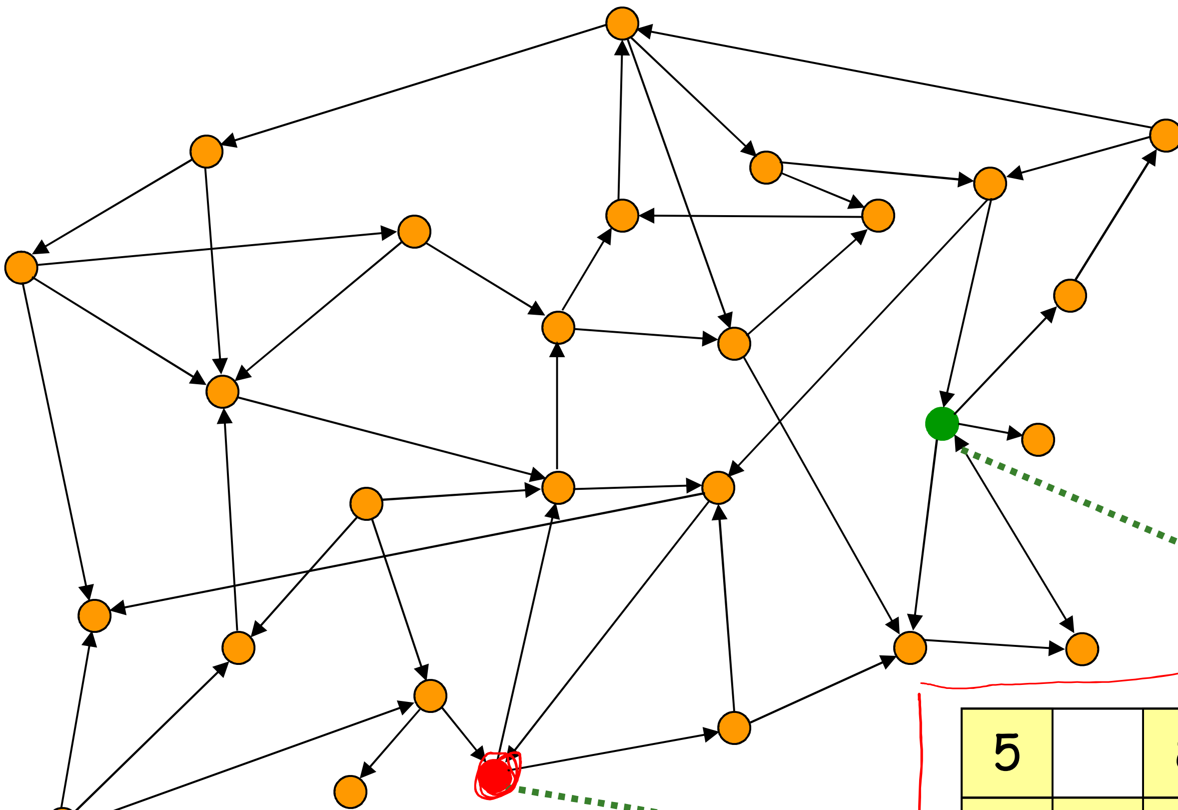
这两个是不能的，
因为不是connected的





1000块钱悬赏解Puzzle但实际上是不可能的因为不是connected

Just to make sure we're clear…



每个Node代表了一种状态

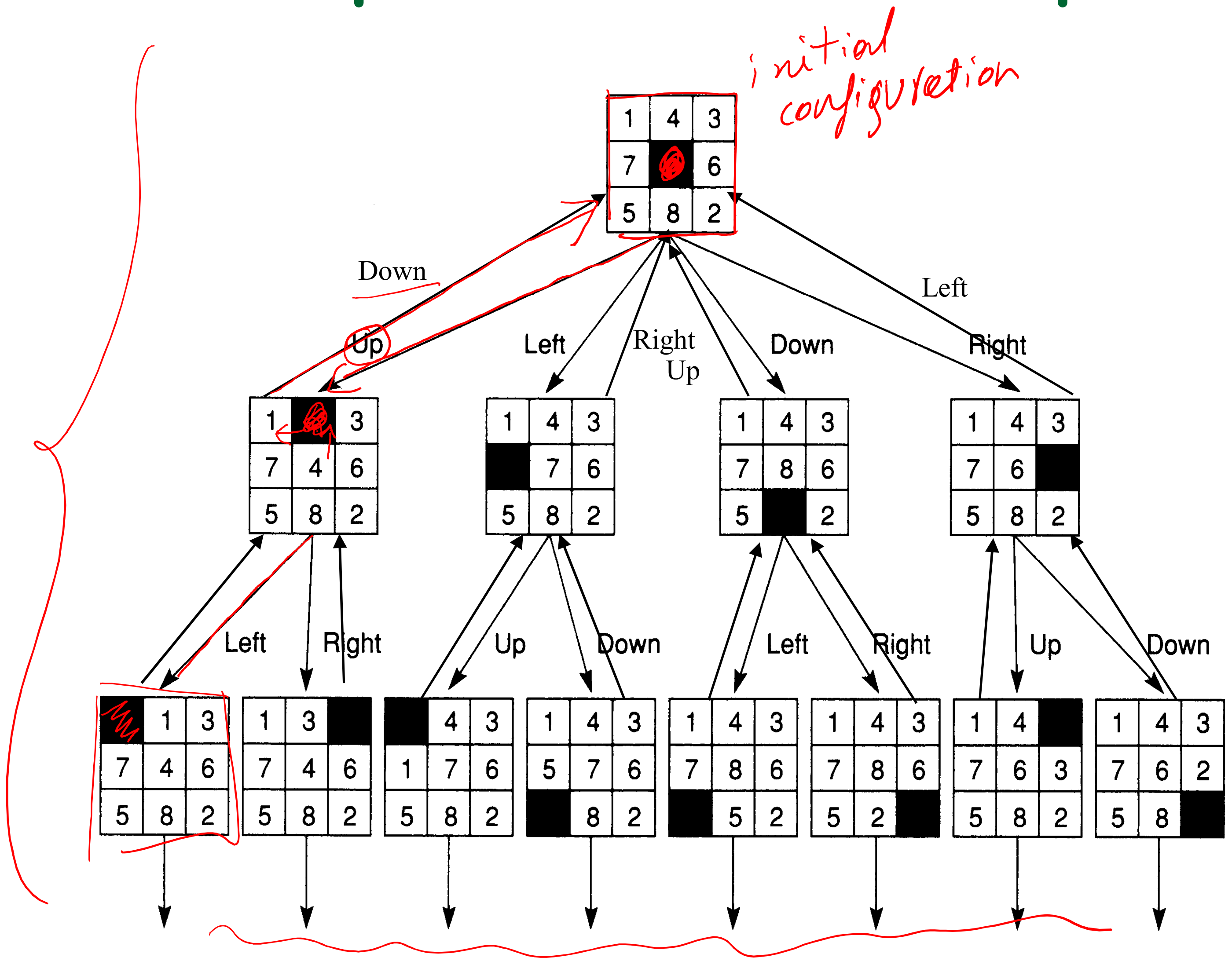
5		8
4	2	1
7	3	6

Initial state

1	2	3
4	5	6
7	8	

Goal state

State Space for the 8-puzzle



source: G. Luger (2005)

Size of state spaces

- For the (n-1)-puzzle:
  - Nb of states:
    - 8-puzzle --> 9! = 362,880 states
    - 15-puzzle --> 16! ~ 2.09×10^{13} states
    - 24-puzzle --> 25! ~ 10^{25} states
  - At 100 millions states/sec:
    - 8-puzzle --> 0.036 sec
    - 15-puzzle --> ~ 55 hours
    - 24-puzzle --> > 10^9 years

无非就是while里加了一步goal，如果得到了，backtrack，得到路径，不行，进行下一个动作

goal(state) → T
→ F

- For real problems:
  - state spaces are way too large to be generated in advance and searched after state space经常容易太大了，很难先生成，再搜索
  - so it is generated dynamically while searching.

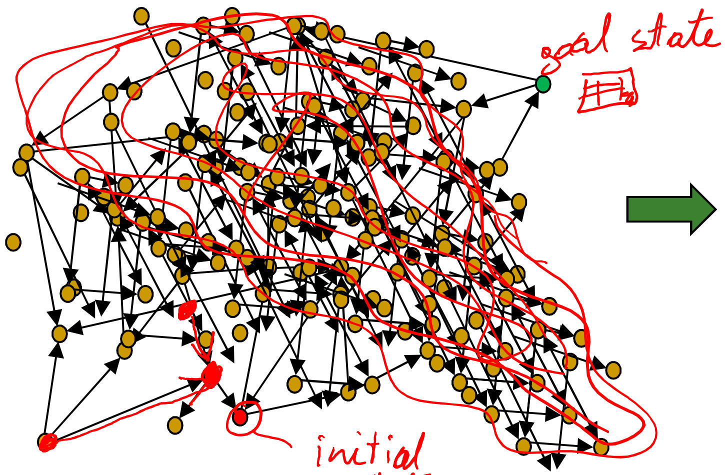
goal(·) → T
→ F

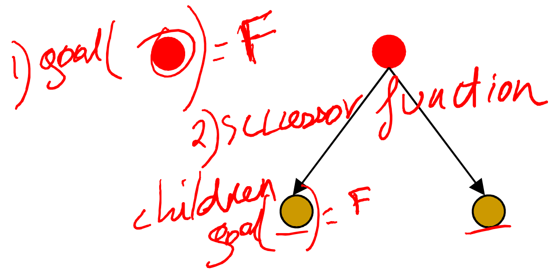
successor

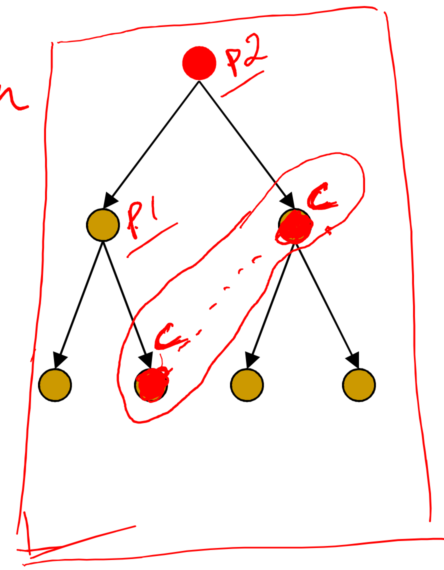
通常动态生成，动态搜索

State Space Graph as a Search Tree

实际上我们在生成一个search tree







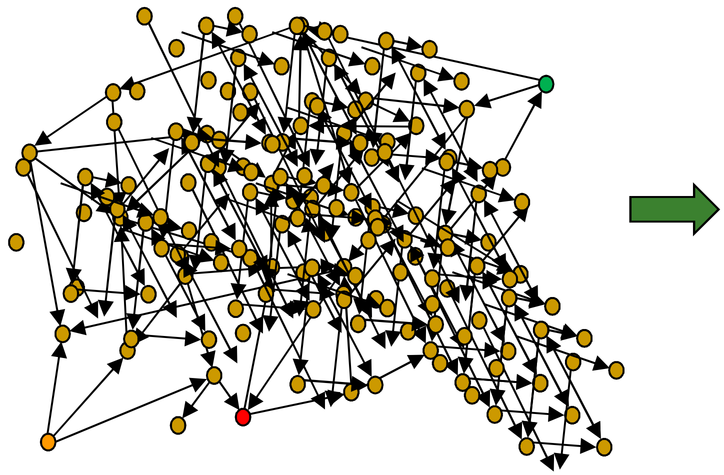
有可能两个点是同一个C，只是有两个不同的parent

- The state space is generated dynamically while searching.
  - we explore a node:
    - if it the goal, stop and trace back the path from the initial node / solution path
    - if it is not the goal, then generating its successors/children and explore these recursively
  - to avoid cycles, the search algorithm will check for duplicate nodes.

为了避免循环我们要避免右上角两个C的情况

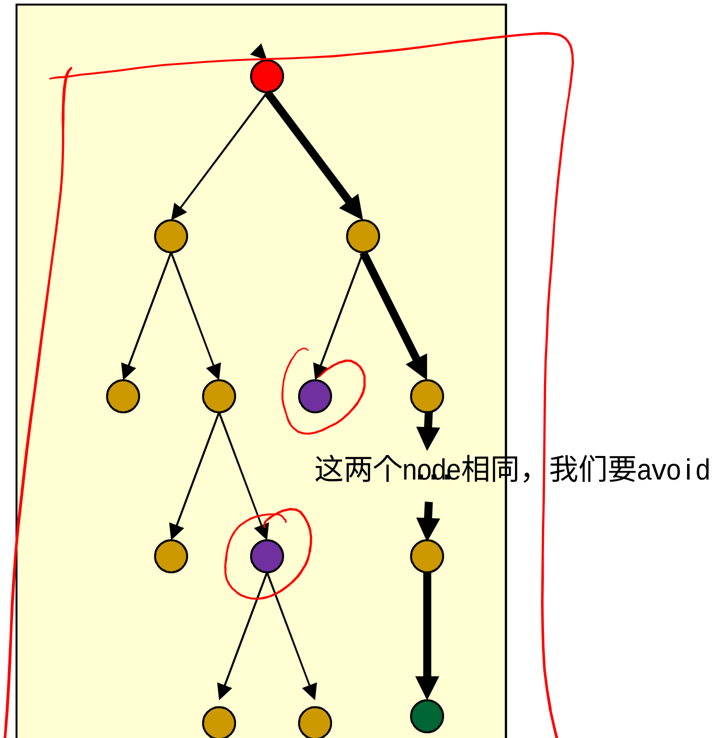
Search tree

State Space Graph as a Search Tree



Theoretical State Space Graph

So now, we just need an efficient search algorithm



Dynamically created Search tree

Today

1. State Space Representation
2. State Space Search
   a) Overview
   b) Uninformed search
      1. Breadth-first and Depth-first
      2. Depth-limited Search
      3. Iterative Deepening
      4. Uniform Cost
   c) Informed search
      1. Intro to Heuristics
      2. Hill climbing
      3. Greedy Best-First Search
      4. Algorithms A & A*
      5. More on Heuristics
   d) Summary

Up Next

1. State Space Representation
2. State Space Search
   a) Overview
   b) Uninformed search
      1. Breadth-first and Depth-first
      2. Depth-limited Search
      3. Iterative Deepening
      4. Uniform Cost
   c) Informed search
      1. Intro to Heuristics
      2. Hill climbing
      3. Greedy Best-First Search
      4. Algorithms A & A*
      5. More on Heuristics
   d) Summary

video #2

COMP 352

novelty